PB82-107053

Users Manual for the Social Survey Computer System

Computer Sciences Corp.
Falls Church, VA

Prepared for

Environmental Protection Agency
Washington, DC

Jul 81

U.S. Department of Commerce
National Technical Information Service
NTIS

PB82-107053

USERS MANUAL FOR THE
SOCIAL SURVEY COMPUTER
SYSTEM

July 1981

U.S. ENVIRONMENTAL PROTECTION AGENCY
Office of Noise Abatement and Control
Washington, D.C. 20460

Under Contract No. 68-01-3840

This report has been approved for general availability. The contents of this report reflect the views of the contractor, who is responsible for the facts and the accuracy of the data presented herein. This report does not necessarily reflect the official views or policy of EPA. This report does not constitute a standard, specification, or regulation.

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA 22161

0272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO.	2.	3. Recipient's Accession No. PB82-107053
4. Title and Subtitle "Users Manual for the Social Survey Computer System			5. Report Date July 1981
			6.
7. Author(s)			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Computer Sciences Corporation 6565 Arlington Boulevard Falls Church, Va. 22046			10. Project/Task/Work Unit No.
			11. Contract(C) or Grant(G) No. (C) (G) 68-01-3840
12. Sponsoring Organization Name and Address U.S. Environmental Protection Agency 401 M Street, S.W. Washington, D.C. 20460			13. Type of Report & Period Covered
			14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

This report was prepared by EPA, Office of Noise Abatement and Control, in support of its function to provide technical assistance to communities. It is one of nine which comprises the Community Noise Assessment Manual. The Manual provides a comprehensive and computerized system for assessing the noise problems of a community and then planning a noise control strategy for its abatement. This Manual consists of a system of computer programs. It is designed to analyze the raw numbers generated by the social survey (the attitudinal survey) and produce from them a series of concise, meaningful reports. The system described in this Manual runs on the UNIVAC 1110 computer under the EXEC 8 Operating System as implemented at the EPA's National Computer Center. The programs are written for the UNIVAC FORTRAN V computer.

17. Document Analysis a. Descriptors

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement: 1< Release to the public	19. Security Class (This Report) Unclassified	21. No. of Pages 225
	20. Security Class (This Page) Unclassified	22. Price

(See ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce

TABLE OF CONTENTS

Section 1 - Introduction

1.1 Purpose 1-1
1.2 Operational Equipment 1-1
1.3 Systems Configuration 1-3
1.4 User Profile 1-3
1.5 Notation Conventions 1-3
1.6 Card Codes 1-3

Section 2 - System Summary

2.1 Overview 2-1
2.1.1 Structure 2-1
2.1.2 Flow 2-4
2.1.3 Using the Manual With the System 2-5

Section 3 - System Operation

3.1 Common System Initiation Procedures 3-1
3.1.1 Detailed Initiation Procedure 3-1
3.1.1.1 Obtain id's 3-1
3.1.1.2 Initiate the System 3-1
3.1.1.3 Initialize the Data File 3-3
3.1.2 System Security 3-3
3.1.2.1 Read Only 3-6
3.1.2.2 Private 3-6
3.1.3 General Error and Recovery Procedures 3-6
3.2 CLEAN 3-7
3.2.1 Security 3-7
3.2.2 Initiation Procedures 3-7
3.2.3 Operational Procedures 3-15
3.2.3.1 Correcting Errors in the Data 3-17
3.2.3.2 Timing 3-20
3.2.4 Error Procedures 3-20
3.2.5 Constraints/Limitations 3-31
3.2.6 Data Base 3-31
3.2.7 Control Language 3-32
3.2.8 Input 3-32
3.2.9 Output 3-32
3.3 VARIANCE 3-33
3.3.1 Security 3-33
3.3.2 Initiation Procedures 3-33
3.3.3 Operation Procedures 3-33
3.3.4 Error and Recovery Procedures 3-36
3.3.5 Constrains/Limitations 3-36
3.3.6 Data Base 3-36
3.3.7 Control Language 3-36
3.3.8 Input 3-36
3.3.9 Output 3-36
3-36

2<

TABLE OF CONTENTS (Cont'd)

Section 3 (Cont'd)

3.4 TAILOR 3-38
3.4.1 Security 3-46
3.4.2 Initiation Procedures 3-46
3.4.3 Operational Procedures 3-46
3.4.4 Error Procedures 3-47
3.4.5 Data Base 3-49
3.4.6 Control Language 3-49
3.4.7 Input 3-49
3.4.8 Output 3-52
3.5 RECODE 1 3-53
3.5.1 Security 3-53
3.5.2 Initiation Procedures 3-53
3.5.3 Operational Procedure 3-53
3.5.4 Error and Recovery Procedures 3-53
3.5.5 Constraints/Limitations 3-53
3.5.6 Data Base 3-53
3.5.7 Control Language 3-53
3.5.8 Input 3-55
3.5.9 Output 3-55
3.5.9.1 Proj-id*SYSFILE 1 3-55
3.5.9.2 Proj-id*ZSCORES 3-55
3.6 RECODE 2 3-56
3.6.1 Security 3-56
3.6.2 Initiation Procedures 3-56
3.6.3 Operational Procedures 3-56
3.6.4 Error and Recovery Procedures 3-56
3.6.5 Constraints/Limitations 3-56
3.6.6 Data Base 3-56
3.6.7 Control Language 3-56
3.6.8 Input 3-56
3.6.9 Output 3-56
3.7 FREQUENCIES 3-58
3.7.1 Security 3-58
3.7.2 Initiation Procedures 3-58
3.7.3 Operational Procedures 3-58
3.7.4 Error and Recovery Procedures 3-58
3.7.5 Contraints/Limitations 3-58
3.7.6 Data Base 3-58
3.7.7 Control Language 3-58
3.7.8 Input 3-58
3.7.9 Output 3-58
3.8 CROSSTABS 3-60
3.8.1 Security 3-60
3.8.2 Initiation Procedures 3-60

3<

TABLE OF CONTENTS (Cont'd)

Section 3 (Cont'd)

3.8.3 Operational Procedures 3-60
3.8.4 Error and Recovery Procedures 3-60
3.8.5 Constraints/Limitations 3-60
3.8.6 Data Base 3-60
3.8.7 Control Language 3-60
3.8.8 Input 3-60
3.8.9 Output 3-60
3.9 REGRESSIONS 3-62
3.9.1 Security 3-62
3.9.2 Initiation Procedures 3-62
3.9.3 Operational Procedures 3-62
3.9.4 Error and Recovery Procedures 3-62
3.9.5 Constraints/Limitations 3-62
3.9.6 Data Base 3-62
3.9.7 Control Language 3-62
3.9.8 Input 3-62
3.9.9 Output 3-62
3.10 Archiving the System's Results 3-64
3.10.1 Security 3-64
3.10.2 Initiation Procedures 3-64
3.10.3 Operational Procedures 3-64
3.10.4 Error and Recovery Procedures 3-66
3.10.5 Constraints/Limitations 3-67
3.10.6 Data Base 3-67
3.10.7 Control Language 3-67
3.10.8 Input 3-67
3.10.9 Output 3-67
3.11 De-archiving the System's Results 3-68
3.11.1 Security 3-68
3.11.2 Initiation Procedure 3-68
3.11.3 Operational Procedure 3-68
3.11.4 Error and Recovery Procedures 3-68
3.11.5 Constraints/Limitations 3-70
3.11.6 Data Base 3-70
3.11.7 Control Language 3-70
3.11.8 Input 3-70
3.11.9 Output 3-70
3.12 Clean-up 3-72
3.12.1 Security 3-72
3.12.2 Initiation Procedures 3-72
3.12.3 Operational Procedures 3-72
3.12.4 Error and Recovery Procedures 3-72
3.12.5 Constraints/Limitations 3-74
3.12.6 Data Base 3-74

4<

TABLE OF CONTENTS (Cont'd)

Section 3 (Cont'd)

3.12.7 Control Language 3-74
3.12.8 Input 3-74
3.12.9 Output 3-74

Appendix A - Information Regarding the EPA'S National Computer Center (NCC)...A-1

Appendix B - Project References - Manuals for the Attitudinal Survey..........B-1

Appendix C - Errors Reported by CLEAN....................................C-1

Appendix D - Runstream Explanations......................................D-1

Appendix E - Univac ECL Summary..E-1

Appendix F - Glossary..F-1

5<

TABLE OF FIGURES

Figure		Page No.
1-1	ASDAS' Relationship to LISTEN	1-2
2-1	Files and Elements Used by the ASDAS Programs	2-2
2-2	Structure and Data Paths in the ASDAS	2-3
3-1	First System Initiation Runstream	3-2
3-2	Second System Initiation Runstream	3-4
3-3	Alternate Second System Initiation Runstream	3-5
3-4	Sample CLEAN Runstream	3-16
3-5	Runstream to Copy proj-id*RAWDATA to proj-id*GOODDATA	3-18
3-6	Sample Edit Runstream	3-21
3-7	Sample Data Cleaning	3-22
3-8	Variable for which VARIANCE Calculates Statistics	3-34
3-9	Sample VARIANCE Runstream	3-35
3-10	Sample of VARIANCE Output	3-37
3-11	Runstream to Run TAILOR on RECODE 1	3-39
3-12	Runstream to Run TAILOR on LABELS	3-40
3-13	Runstream to Run TAILOR on RECODE 2	3-41
3-14	Runstream to Run TAILOR on FREQUENCIES	3-43
3-15	Runstream to Run TAILOR on CROSSTABS	3-44
3-16	Runstream to Run TAILOR on REGRESSIONS	3-45
3-17	Sample Output from TAILOR Program	3-48
3-18	TAILOR Insertion of Acoustical Data into RECODE 2	3-50
3-19	Sample RECODE 1 Runstream	3-54
3-20	Sample RECODE 2 Runstream	3-57
3-21	Sample FREQUENCIES Runstream	3-59
3-22	Sample CROSSTABS Runstream	3-61
3-23	Sample REGRESSIONS Runstream	3-63
3-24	Runstream to Achieve ASDAS Results	3-65
3-25	Runstream to De-achieve ASDAS Results	3-69
3-26	Sample Runstream to Delete LISTEN System	3-73
A-1	Useful NCC Phone Numbers	A-3
A-2	Phone Numbers for NCC Remote Batch Entry Access	A-4
A-3	Numbers to Call to Gain Access to NCC Time Sharing	A-5
E-1	Notation Conventions	E-18

6<

LIST OF ILLUSTRATIONS

Number	Title	Page No.
3-1	Questionnaire Data Card: Layouts and Legal Values	3-8
3-2	The CLEAN Initialization Card	3-19
A-1	NCC Operations Schedule	A-6
A-2	NCC Remote Site ID's	A-7
D-1	Comparison of Fieldator and EBDIC	D-2
E-1	ECL Summary	E-3

7<

SECTION 1 - INTRODUCTION

1.1 PURPOSE

The Attitudinal Survey Data Analysis System (ASDAS) is a system of computer programs. It is designed to analyze the raw numbers generated by a sociological survey (the Community Noise Assessment Social Survey) and produce from them a series of concise, meaningful reports. The system is administered by the State and Local Programs Division of the EPA's Office of Noise Abatement and Control (ONAC). In conjunction with the Acoustical Data Reduction and Noise Optimization (NOIZOP) computer programs, ASDAS makes up the data processing end of a larger system of data collection and data processing protocols. This larger system is called LISTEN, an acronym for Local Information System to Evaluate Noise. (Strictly speaking, LISTEN refers only to the data processing end of this system, but the term is often used to refer the system as a whole). LISTEN enables a community to determine the most effective combination of noise control measures to employ, given:

- The nature and distribution of noise sources within the community
- The extent and manner in which noise affects the community's residents
- The budgetary limits imposed on the community's planners. Figure 1-1 shows ASDAS' relationship to the rest of LISTEN.

1.2 OPERATIONAL ENVIRONMENT

ASDAS runs on the UNIVAC 1110 computer under the EXEC 8 operating system as implemented at the EPA's National Computer Center (NCC), Research Triangle Park, North Carolina. The following system processors are also required:

- SPSS (Statistical Package for the Social Sciences) Version 7.2
- The University of Maryland text editor (identified as US*ER.ED on the NCC system)

All FORTRAN programs were compiled using the UNIVAC FORTRAN V (release 4R1) compiler.

8<

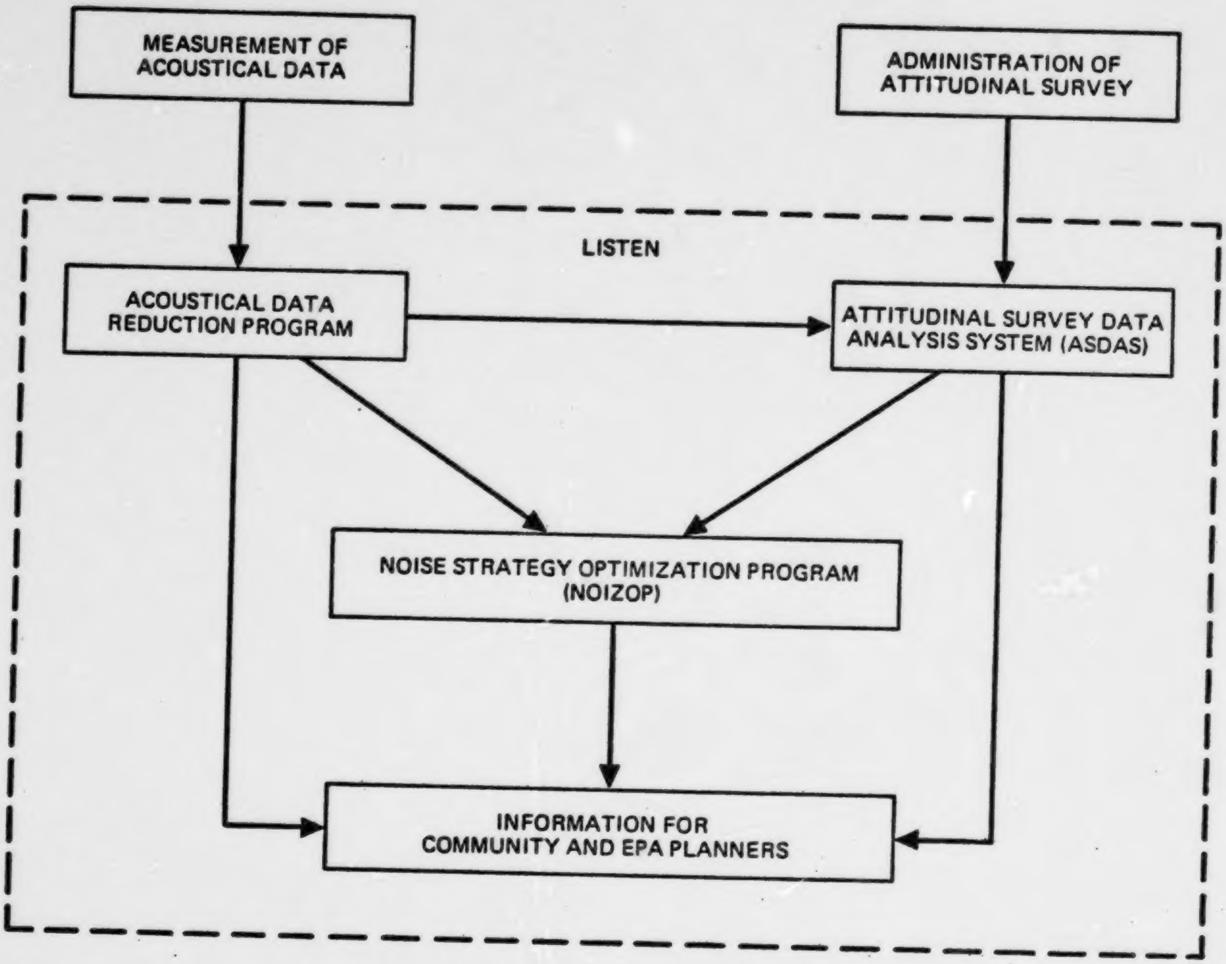

MEASUREMENT OF ACOUSTICAL DATA
ADMINISTRATION OF ATTITUDINAL SURVEY
LISTEN
ACOUSTICAL DATA REDUCTION PROGRAM
ATTITUDINAL SURVEY DATA ANALYSIS SYSTEM (ASDAS)
NOISE STRATEGY OPTIMIZATION PROGRAM (NOIZOP)
INFORMATION FOR COMMUNITY AND EPA PLANNERS

Figure 1-1

ASDAS Relationship to Listen

9<

1.3 SYSTEMS CONFIGURATION

It is anticipated that the system will be run at a remote site. Therefore, in addition to equipment and software listed in 1.2 above, a remote card reader and line printer will be needed. Information on interfacing this equipment with the UNIVAC 1110 computer is available from NCC user support personnel. Appendix A contains a list of NCC contacts.

1.4 USER PROFILE

It is intended that this system be run by EPA or EPA contractor data entry personnel. No programming experience should be necessary to use this manual and the system correctly. It will be advisable, however, to have an experienced programmer available in case any questions arise.

1.5 NOTATION CONVENTIONS

This manual follows the standard practice of presenting all user supplied information in lower case when shown in a runstream example. Information that is standard for all users and that must be punched as given in the example is presented in UPPER CASE.

File names under the UNIVAC EXEC 8 operating system are composed of two parts - a qualifier and a name, and are written in the form qualifier name. If the qualifier is omitted it is assumed to be the same as the proj-id.

1.6 CARD CODES

See Appendix D for information on possible card code conflicts.

10<

SECTION 2 - SYSTEM SUMMARY

2.1 OVERVIEW

2.1.1 STRUCTURE

The ASDAS consists of eight programs which perform four tasks utilizing seven data files. One data file serves as input to the program; six contain output. In addition, several intermediate files are produced by the system. The programs are named:

CLEAN	(FORTRAN)
VARIANCE	"
TAILOR	"
RECODE1	(SPSS)
RECODE2	"
FREQUENCIES	"
CROSSTABS	"
REGRESSIONS	"

The data files are listed in Figure 2-1.

The tasks accomplished by ASDAS are:

- Data cleaning,
- Variance analysis,
- SPSS program generation,
- Statistical analysis.

The data files used contain:

- Raw input data from the questionnaire,
- The output from the data cleaning, variance analysis and statistical analysis portions of the system.

The raw data must be on a disk file that is created from punched cards provided by the community which administered the survey. All other data files are created by the system and reside on disk. Figure 2-2 shows the flow of data through the system.

```
LISTEN*ATTITUDINAL
                  .RECODE1
                  .LABELS
                  .RECODE2
                  .FREQUENCIES
                  .CROSSTABS
                  .REGRESSIONS
                  .CLEAN
                  .VARIANCE
                  .TAILOR
proj-id*ATTITUDINAL
                  .RECODE1
                  .LABELS
                  .RECODE2
                  .FREQUENCIES
                  .CROSSTABS
                  .REGRESSIONS
```

DATA FILES

```
proj-id*RAWDATA.
proj-id*GOODDATA.
proj-id*TAILORINPUT.
proj-id*TAILOROUTPUT.
proj-id*VARIANOUTPUT.
proj-id*SYSFILE1.
proj-id*ZSCORES.
proj-id*SYSFILE2.
proj-id*FREQOUT.
proj-id*XTABSOUT.
proj-id*REGRESSOUT.
```

Figure 2-1

Files and Elements Used by the ASDAS Programs

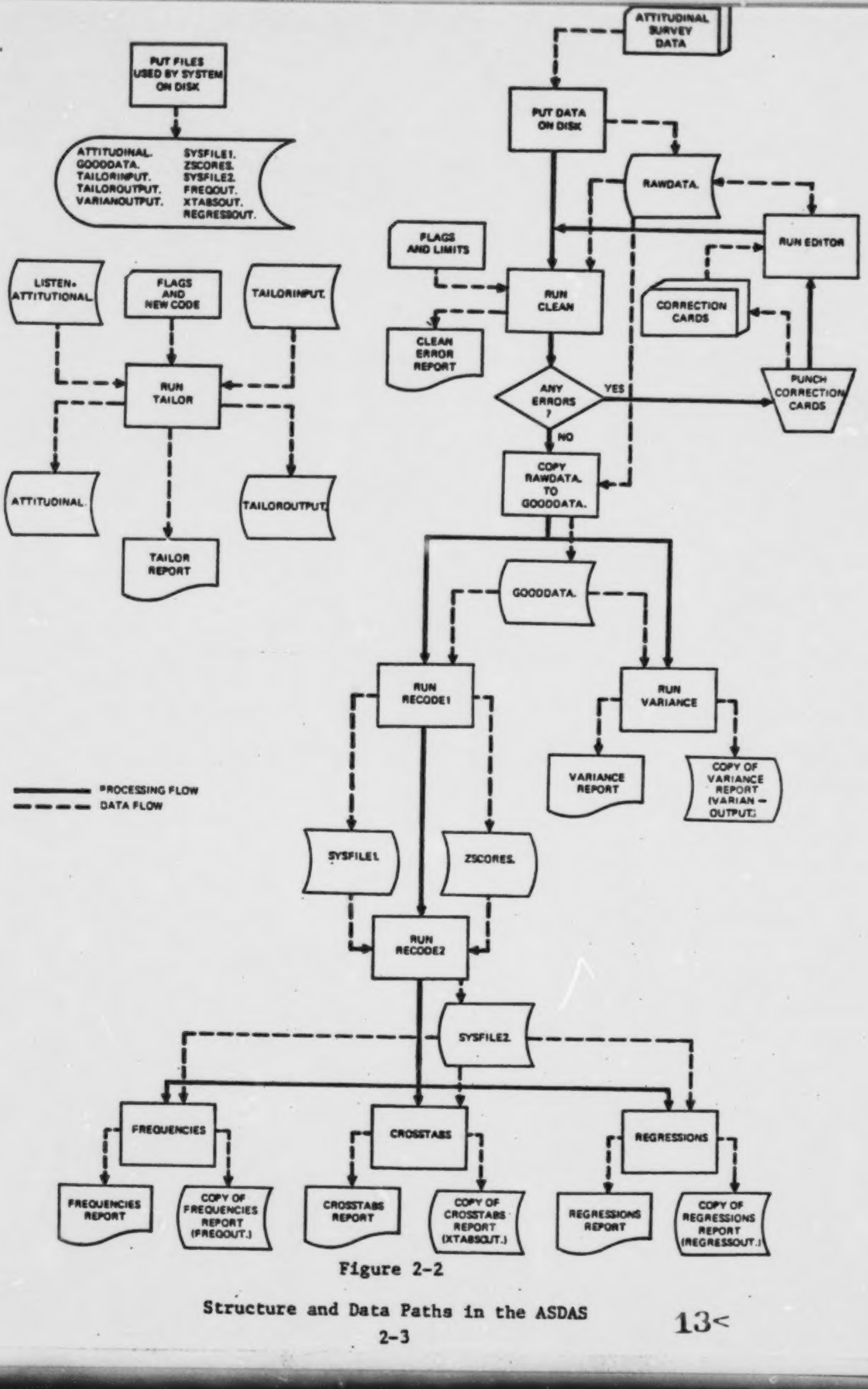

PUT FILES USED BY SYSTEM ON DISK
ATTITUDINAL.
GOODDATA.
TAILORINPUT.
TAILOROUTPUT.
VARIANOUTPUT.
SYSFILE1.
ZSCORES.
SYSFILE2.
FREQOUT.
XTABSOUT.
REGRESSOUT.
LISTEN. ATTITUTIONAL.
FLAGS AND NEW CODE
TAILORINPUT.
RUN TAILOR
ATTITUDINAL.
TAILOROUTPUT.
TAILOR REPORT
ATTITUDINAL SURVEY DATA
PUT DATA ON DISK
RAWDATA.
RUN EDITOR
FLAGS AND LIMITS
RUN CLEAN
CLEAN ERROR REPORT
CORRECTION CARDS
ANY ERRORS ?
YES
NO
PUNCH CORRECTION CARDS
COPY RAWDATA. TO GOODDATA.
GOODDATA.
RUN RECODE1
RUN VARIANCE
VARIANCE REPORT
COPY OF VARIANCE REPORT (VARIAN - OUTPUT.)
PROCESSING FLOW
DATA FLOW
SYSFILE1.
ZSCORES.
RUN RECODE2
SYSFILE2.
FREQUENCIES
CROSSTABS
REGRESSIONS
FREQUENCIES REPORT
COPY OF FREQUENCIES REPORT (FREQOUT.)
CROSSTABS REPORT
COPY OF CROSSTABS REPORT (XTABSOUT.)
REGRESSIONS REPORT
COPY OF REGRESSIONS REPORT (REGRESSOUT.)

Figure 2-2

Structure and Data Paths in the ASDAS

2.1.2 FLOW

In the first step, data cleaning, the raw data from the questionnaire are fed to the CLEAN program which reports all detectable errors. These errors are corrected by the user and the modified data must then be submitted to the CLEAN program for more checking. This process is repeated until no more errors are found by CLEAN.

Once the data has been corrected, the user begins the second and third tasks. The second task, determination of the error introduced by computer sampling, is performed by the VARIANCE program. This program calculates a variety of statistics, most importantly the variance, for a number of variables. From the variance we can determine the error induced by the sampling method used (stratified cluster sampling).

The third task is the selection of a configuration for each of the SPSS statistical analysis programs. There are four possible configurations depending upon whether or not a community has a noise control program and whether or not it has conducted an acoustical survey. The program used to select the proper configuration is called TAILOR.

The fourth and final task is statistical analysis. Five programs are used to perform this analysis:

- RECODE1
- RECODE2
- FREQUENCIES
- CROSSTABS
- REGRESSIONS.

RECODE1 and RECODE2 change the questionnaire data into a form useable by the other three programs. FREQUENCIES generates frequency distributions for selected survey questions. CROSSTABS produces contingency tables from the data, while REGRESSIONS performs a multiple regression analysis on the data. They are altered by the TAILOR program described above.

The user may wish to save the results produced by the system on magnetic tape. Instructions on how to do this are included in this manual.

2.1.3 USING THE MANUAL WITH THE SYSTEM

This manual, and the system as a whole, have been designed to minimize the amount of specialized knowledge necessary. The system is intended to operate in batch mode and in a "hands-off" fashion, with the user merely supplying data to be inserted into previously defined runstreams. Because of this, it should be possible for data processing personnel from the communities involved to run the ASDAS, provided they have the hardware required to interface with the NCC computer.

Users who wish to use timesharing to submit their batch runs will need to have some familiarity with the UNIVAC EXEC 8 operating system. Appendix E describes some common EXEC 8 commands.

The user who wishes a thorough understanding of the LISTEN system will find it useful to review the User's Manuals for the other computer programs associated with LISTEN: the Acoustical Data Reduction and NOIZOP programs, but this is not a prerequisite to running the ASDAS.

15<

SECTION 3 - SYSTEM OPERATION

3.1 COMMON SYSTEM INITIATION PROCEDURES

Before the ASDAS can be employed, the user must perform the following steps:

- Obtain user identifiers (id's)
- Initiate the system
- Initialize the data file to be used.

3.1.1 DETAILED INITIATION PROCEDURE

3.1.1.1 Obtain id's

The user must begin initiation procedures by obtaining a valid user identification (user-id), account number (acc-num), site identification (site-id), and password from the State and Local Programs Division's data processing coordinator. These identifiers will be used to run jobstreams throughout the system's operation.

The user must also choose a project identifier (proj-id). This identifier must be less than 12 characters long. It is suggested that the name of the city whose data is being analyzed be used as the project identifier. For example, SPOKANE might be used as the proj-id for Spokane, WA data analysis.

3.1.1.2 Initiate the System

To initiate the system, it is necessary to create a number of files for the system's later use. Follow this procedure:

- Refer to the runstream shown in Figure 3-1.
- Substitute the values that apply to the user for the placeholders 'user-id', 'acc-num', 'proj-id' and 'password'
- Punch the runstream on cards
- Submit the cards

Appendix D contains line by line descriptions of every example runstream. Here we only note that at the conclusion of the run the following files will exist on disk:

```
@RUN,R/RT INIT1/25/0,acc-num/user-id,proj-id,S030,20
@PASSWD password
@DELETE,C ATTITUDINAL.
@DELETE,C GOODDATA.
@DELETE,C TAILORINPUT.
@DELETE,C TAILOROUTPUT.
@DELETE,C VARIANOUTPUT.
@DELETE,C SYSFILE1.
@DELETE,C ZSCORES.
@DELETE,C SYSFILE2.
@DELETE,C FREQOUT.
@DELETE,C XTABSOUT.
@DELETE,C REGRESSOUT.
@CAT ATTITUDINAL.
@CAT GOODDATA.
@CAT TAILORINPUT.
@CAT TAILOROUTPUT.
@CAT VARIANOUTPUT.
@CAT SYSFILE1.
@CAT ZSCORES.
@CAT SYSFILE2.
@CAT FREQOUT.
@CAT XTABSOUT.
@CAT REGRESSOUT.
@FIN
```

Figure 3-1

First System Initiation Runstream

proj-id*ATTITUDINAL.
proj-id*GOODDATA.
proj-id*TAILORINPUT.
proj-idTAILOROUTPUT.
proj-id*VARIANOUTPUT.
proj-id*SYSFILE1.
proj-id*ZSCORES.
pron-id*SYSFILE2.
proj-id*FREQOUT.
proj-id*XTABSOUT.
proj-id*REGRESSOUT.

3.1.1.3 Initialize the Data File

The next step is to transfer the questionnaire data from punched cards to disk storage. The file proj-id*RAWDATA. will contain the data. To make the transfer, use the runstream in Figure 3-2 or Figure 3-3 following this procedure:

- Punch the runstream on cards, substituting for the placeholders 'user-id', 'acc-num', 'proj-id' and 'password'.
- Place the questionnaire data cards in the runstream at the point indicated in the figure.
- Submit the run.

If successfully run, there will exist on disk a file named proj-id*RAWDATA. which contains the questionnaire data.

3.1.2 SYSTEM SECURITY

The Attitudinal Survey Data Analysis System (ASDAS) assumes that no attempt will be made to maliciously damage data contained therein. Because of this assumption, system security consists, for the most part, of ensuring against accidental destruction of data or files. This is accomplished by making all files

- Read-only
- Private (wherever possible)

```
@RUN,R/RT INIT2/25/0,acc-num/user-id,proj-id,004,150
@PASSWD password
@DELETE,C RAWDATA.
@FILE,CR proj-id*RAWDATA.

       .
       .
       .
       questionnaire data
       .
       .
       .
@ENDF
@FIN
```

Figure 3-2

Second System Initiation Runstream

```
@RUN,R/RT INIT2A/25/0,acc-num/user-id,proj-id,004,150
@DELETE, C RAWDATA.
@DATA, I RAWDATA.,,SENTNL
        .
        .
        .
        questionnaire data
        .
        .
        .
@END SENTNL
@FIN
```

For use in the case where the run is to be entered to the system via the @START command from a demand terminal.

Figure 3-3

Alternate Second System Initiation Runstream

20<

3.1.2.1 Read Only

Read only, as its name suggests, means that a file can be accessed for read operations alone. Writes to read-only files will fail, which provides a measure of data protection.

3.1.2.2 Private

Under EXEC 8 as presently configured at the National Computer Center, designating a file as private prevents a user from accessing it unless he is running under the same project-id as the file's creator. This provides additional protection against accidental damage to the file and some protection against deliberate destruction or misuse. The operating system does not check to see if the accessing user and the creating user have matching user-ids and/or account numbers. This limits the protection that private files provide against a malicious user.

The use of read and write keys was rejected as being too cumbersome and not justified by the data's sensitivity except in the case of the city archive tape (see Section 3.10). A write key was used for that tape to prevent the data from being accidentally destroyed.

3.1.3 GENERAL ERROR AND RECOVERY PROCEDURES

ASDAS is subject to few problems. The most likely are:

- Hardware or operating system failure during a run
- File rollout

A hardware or operating system failure will cause ASDAS to leave files in incomplete and incorrect forms, but this is the only damage it will cause. The runstreams are designed so that all that the user need do to fix the system is simply run the interrupted job again.

File rollout is when a file is moved from disk to a backup tape because of lack of use. If this happens to SPSS or another system processor and a runstream attempts to access this processor, the vague error message 'FILE ERROR' will be generated and the run aborted. The attempt to use the processor is noted by the operating system which will then return the processor to disk. This takes 10 to 30 minutes. The user should resubmit the job after that period has elapsed.

21<

No other errors are expected to occur. If one does, the user should save the listing of the run in which it was encountered, rerun the job, and, if it occurs a second time, take both listings to a systems programmer for analysis.

The user will note that the system and its runstreams are constructed in such a way that the user may after running the system to a certain point go back to any previous point and resume execution of the system there. This minimizes the effect of any mistake made in running the system since only the part of the system in which the error was made and the parts following it need be rerun.

3.2 CLEAN

The purpose of CLEAN is to:

- Scan the raw data from the attitudinal survey
- Compare the values found in the data against the valid values for each question.
- Check the data's self-consistency. (If, for example, a respondent who is reported to be 19 years old is also reported to have 21 years of education, CLEAN will detect the error.)
- When an illegal value is found, print a message describing the type of error, the question(s) affected and the card columns that the erroneous data are punched in.

Table 3-1 is a summary of the fields on the various data cards and the values CLEAN will pass without comment.

3.2.1 SECURITY (See subsection 3.1.1)

3.2.2 INITIALIZATION PROCEDURES

Two steps must be taken before CLEAN can be run:

- The file proj-id*RAWDATA. must be made available,
- An initialization card must be prepared.

The file will be available if the second system initialization run has completed successfully.

22<

Table 3-1

Questionnaire Data Cards:
Layouts and Legal Values

CARD ONE

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID NO.*	1-5	00001	01500		Cannot = previous CARD 1 ID #
Card	6-7			01	
Noise Zone*	8-9	01	16		
Area*	10	1	7		
Cluster*	11-13	001	100		
City*	14-16			Community Dependent	
Selection Label	17-18	00	12		
Number of Calls	19	1	9		
Interview Date*	20-21	01	Community Dependent		
Total Number of Adult Residents	22	1	8		
Respondent Number on List	23	1	8		Must be < or = to TOT. NO. ADULT RESIDENTS
Sex of Respondent	24			1, 2	Must correspond to CARD 4, B5
Age of Respondent	25-26	18	99		Must correspond to CARD 4, Q22
	27-80				Blanks Only

*With flexible data ranges set at recommended values.

CARD TWO

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID No.	1-5	00001	01500		Must match CARD 1 ID
Card	6-7			02	
Q1	8-9	01	90	95, 98, 99	
Q2	10	1	5	8, 9	
Q3 (A-C)	11-12	00	32	98, 99	
	13-14				
	15-16				
Q4 (A-C)	17-19	101	111	0, 199	
	20-22	201	214	299	
	23-25	301	310	399	
		401	404	499	
		501	506	599	
		601	603	699	
		701	704	799	
		801	807	899	
		901	906	950, 975 998, 999	
Q5 (A-H)	26,27	1	5	8,9	
	28,29				
	30,31				
	32,33				

Continued Next Page

CARD TWO Continued

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
Q6(A-K)	34,35	1	5	8,9	
	36,37				
	38,39				
	40,41				
	42,43				
	44				
Q7	45	1	5	8,9	
Q8	46	1	4	8,9	
Q9	47	1	9		
Q10(A-D)	48,49	1	5	8,9	
	50,51				
Q11(A-F)	52,53			0,1,8,9	
	54,55				
	56,57				
Q12	58	1	3	8,9	
Q13(A-E)	59,60	1	5	8,9	
	61,62				
	63				
Q14	64	1	5	8,9	
Q15(A-G)	65,66			0,1,8,9	
	67,68				
	69,70				

Continued Next Page

CARD TWO Continued

FIELD NAME	Cols.	Min.	Mac.	Fixed Val.	Additional Constraints
Q15(A-G)(Cont.)	71				
Q16(A-C)	72,73			0,1,8,9	
	74				
	75-80				Blanks Only

CARD THREE

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID No.	1-5	00001	01500		Must = CARD 1 ID NO.
Card	6-7			03	
Q17(A-T)	8,9,10	1	9		
	11,12				
	13,14				
	15,16				
	17,18				
	19,20				
	21,22				
	23,24				
	25,26				
	27				
Q17(U-V)	28,29	00	99		
	30,31				
Q18(A-F)	32,33			0,1,8,9*	
	34,35			8+	
	36,37				
Q19	38			1,2,8,9*+	
Q19A(A)	39	0	2	8,9*	If zero then Q19 must = 2. If 1,2 8, or 9 then Q19 must = 1
		1	4+		
Q19A(B-D)	40,41	0	2	8,9*	11
	42			8+	Must = 8 if 'with' is indicated

27<

*Without Noise Control Program (CLTPRG=1)
+With Noise Control Program (CTLPRG=2)

Continued Next Page

CARD THREE Continued

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
Q20	43	1	9		
Q21(A-H)	44,45			0,1,8,9	
	46,47				
	48,49				
	50,51				
	52-80				Blanks Only

CARD FOUR

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID NO.	1-5	00001	01500		Must = CARD 1 ID #
Card	6-7			04	
Q22	8-9	18	99		
Q23	10-11	00	21	98,99	
Q24A	12-13	01	07	80,98,99	
Q25A	14-15	00	15	98,99	Must = 0 if Q24A = 80, Must not = 0 if Q24A Not = 80
Q26	16-17	01	10	98,99	
Q27	18	1	9		
Q28	19	1	4	8,9	
Q29	20-21	01	99		
Time	22-23	00	99		
B1	24			1,2,9	
B2	25	0	9		
B3	26-28	000	999		
B4	29			1,2,9	
B5	30			1,2,9	
B6	31	1	3	9	
	32-80				Blanks Only

An initialization card is necessary to allow a degree of flexibility in the administration of the survey without modifying CLEAN for each special case.

The first twelve fields of the initiation card, each five columns wide, are used to set the valid value, or range of values, for the community dependent variables found on card one of every case. Table 3-2 shows the format of the initialization card and gives standard values for a number of the card's fields. These standard values are taken from Reference 1, Appendix B. The HI-LO variable naming convention indicates a range of values. The four city codes are fixed values, but may all be different.

The last two fields of the initialization card have values for program control variables entered in them. These variables are:

- MAXERR
- CTLPRG

MAXERR fixes the limit for the number of major errors CLEAN can encounter in the data without terminating early. This feature saves the time and paper that would otherwise be wasted if, as is possible, one or two misplaced cards caused the rest of the survey data to appear to be in error.

CTLPRG is a flag that indicates which attitudinal survey was used to produce the data. If CTLPRG=1, the version suited for communities without noise control programs was used. CTLPRG=2 means the survey suited for communities with noise control programs was used.

3.2.3 OPERATIONAL PROCEDURES

An example runstream for CLEAN is shown in Figure 3-4. In this runstream, the community dependent variables are all set to their standard values. Values for the city codes CTY1, CTY2, CTY3, and CTY4 of 88, for the maximum interview date of 60 and for MAXERR of 9 were chosen arbitrarily. The community is without a noise control program.

User should:

- Submit a similar runstream to run CLEAN on his data, making sure to first,
- Change the values of the community dependent variables and the control flags on the initialization card to the correct ones for the community involved and

```
@RUN,R/RT CLEAN/13/0,acc-num/user-id,proj-id,004,500
@PASSWD password
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A RAWDATA.
@USE 10,RAWDATA.
@XQT LISTEN*ATTITUDINAL.CLEAN
00001 01500000010001703001000070000100100000880008800088000880006000009
@FIN
```

Figure 3-4

Sample CLEAN Runstream

- Use the appropriate values for user-id, acc-num, proj-id and password on the @RUN card

The output from the run will consist of a printout of runstream commands, operating system replies to those commands, output from the CLEAN program and billing information. It will automatically be sent to the line printer at the site where the job was entered.

CLEAN's output can take three forms:

- Complete execution, no errors detected;
- Complete execution, some errors detected;
- Incomplete execution, errors detected.

The latter two cases mean that changes are necessary to the data contained in proj-id*RAWDATA. Section 3.2.3.1 will explain how to make these changes. If the run is successful it executes to completion with no errors detected. The user should then submit the runstream shown in Figure 3-5. This runstream copies the file proj-id*RAWDATA. to proj-id*GOODDATA.

3.2.3.1 Correcting Errors in the Data

A CLEAN run that indicates that errors exist in the data file can end in three ways:

- Normally, after all cases are checked;
- Early, after MAXERR has been exceeded;
- Early, if NCARD, the card number punched in columns 6 and 7 of each card, is ever less than or equal to 0 or greater than or equal to 5.

MAXERR is defined in 3.2.2 above and in Table 3-2.

The user's response will be the same in all three cases, to correct the data. They will be corrected by:

- Reviewing CLEAN's output, making note of record number (card, or in a disk file, card image number), column number and correct value for each error detected by CLEAN.
- Punching two cards with the format shown below for each error CLEAN detects.

32<

```
@RUN,R/RT CHANGE/25/0,acc-num/user-id,proj-id,S030,50
@PASSWD password
@CHG,Z GOODDATA.
@ERS GOODDATA.
@ASG,A RAWDATA.
@ASG,A GOODDATA.
@COPY RAWDATA.,GOODDATA.
@FREE GOODDATA.
@CHG,V GOODDATA.
@FIN
```

Figure 3-5

Runstream to Copy proj-id*RAWDATA. to proj-id*GOODDATA.

Table 3-2

THE CLEAN INITIALIZATION CARD

FIELD NAME	COLS	STANDARD VALUE	REMARKS
LOID	1-5	00001	
	6	blank	
HIID	7-11	01500	
LONZ	12-16	00001	
HINZ	17-21	00017	
LOAR	22-26	00001	
HIAR	27-31	00007	
LOCLUS	32-36	00001	
HICLUS	37-41	00100	
CTY1	42-46		If less than four city codes are used, set the extra fields to one of the codes that is used
CTY2	47-51		
CTY3	52-56		
CTY3	57-61		
MAX INTERVIEW DATE	62-66		Community dependent
MAXERR	67-71	00009	
	72	blank	
CTLPRGS	73		1=without 2=with (a community noise control program)
	74-80	blank	

- Placing all such pairs of cards after the

  @US*ER.ED RAWDATA

  card in the runstream shown in Figure 3-6.

- Submitting the runstream after substituting for 'user-id', 'acc-num', proj-id' and 'password'.

The format for the error correction cards is:

```
rec-num
R,col val
```

Where rec-num is to be replaced by the number of the record in which the error occurred (use the leftmost column for multi-column variables) and val by the correct value of the variable. Multi-column variables must have <u>all</u> their places filled (e.g., a three column variable must have three digits typed). The cards are punched beginning in column 1. Only 1 space is left between col and val.

Figure 3-7 shows the complete process, from the first CLEAN run to error correction to the final CLEAN run to copying proj-id*RAWDATA, into proj-id*GOODDATA.

3.2.3.2 <u>Timing</u>

Roughly 50 seconds of CPU time are required for CLEAN to process the data from 700 questionnaires. The maximum memory CLEAN uses is 13 kWords. The timing for the correction runs is uncertain because it depends upon the number of corrections and where they are made. It is strongly suggested that the corrections be made in order of line number; this saves a significant amount of computer time.

3.2.4 ERROR PROCEDURES

It is not likely that CLEAN will fail. If it does, it will probably be for one of these reasons:

- An operating system or hardware failure at the NCC;
- System initiation runs 1 and 2 did not run correctly.

The two types of failure should be easily distinguishable.

The user's response to the first kind of failure is to resubmit the run. The second type requires that system initiation runs 1 and 2 be resubmitted and CLEAN submitted following their successful completion.

```
@RUN,R/RT EDIT/35/0,acc-num/user-id,proj-id,003,300
@PASSWD password
@CHG,Z RAWDATA.
@ASG,A RAWDATA.
@US*ER.ED RAWDATA.
            .
            .
            .
    user punched cards go here
            .
            .
            .
EXIT
@FREE RAWDATA.
@CHG,V RAWDATA.
@FIN
```

Figure 3-6

Sample Edit Runstream

36<

```
*** ATTITUDINAL SURVEY DATA CLEANING BEGINS ***
PROGRAM WRITTEN BY:  DR. RAGU SINGH, CSC
LAST EDIT BY:  STEVEN G. EAGLESON, CSC, FALLS CHURCH, VA.
        ********VERSION 1.0(0)*********


        FLEXIBLE DATA RANGES:
        LOID=     1  HIID= 9999
        LONZ=     0  HINZ=   30
        LOAR=     0  HIAR=   40
>       LOCLUS=    1  HICLUS=  400
        CITIES=   88      4       99       99
        MAX INT DATE=   60
        MAX NO OF ERRORS=   19

        WITH COMMUNITY NOISE PROGRAM
        AFFECTS Q18, Q19 Q19A, Q21


 ERROR: RECORD NO.=      11, CASE NO.=       3, ID NO.= 8411, CARD NO.= 3
          0841103433 7557677277 7777757000 0888888108 8818111111 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      15, CASE NO.=       4, ID NO.= 8412, CARD NO.= 3
          0841203477 7737777477 7777777000 0888888108 8831118110 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      19, CASE NO.=       5, ID NO.= 8415, CARD NO.= 3
          0841503667 7777777677 7777777000 0888888108 8851111111 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      23, CASE NO.=       6, ID NO.= 8501, CARD NO.= 3
          0850103454 7377767347 7777777000 0888888108 8861111111 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      27, CASE NO.=       7, ID NO.= 8502, CARD NO.= 3
          0850203777 7777777777 7777777000 0888888108 8871101110 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39
```

Figure 3-7 Sample Data Cleaning

Initial CLEAN Run

(continued)

```
 ERROR: RECORD NO.=      30, CASE NO.=       8, ID NO.= 8503, CARD NO.= 2
          0850302013 0111121010 0000011145 5348241173 2133336111 1000001111 211500
0000 1000000000
          Q6F  OUT OF RANGE: =7
          COLUMN 39

 ERROR: RECORD NO.=      35, CASE NO.=       9, ID NO.= 8506, CARD NO.= 3
          0850603667 7667777677 7777777000 0888888108 8881101111 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      36, CASE NO.=       9, ID NO.= 8506, CARD NO.= 4
          0850604281 4010803120 2252103012 1000000000 0000000000 0000000000 000000
0000 0000000000
          TOT.NO. ADULT RESIDENTS, COL 22,  OF CARD 1 LARGER THAN Q29, COLS 20-21,
 OF CARD 4
          IMAGE OF PREVIOUS CARD 1:
          0850601118 0850040122 1312280000 0000000000 0000000000 0000000000 000000
0000 0000000000

 ERROR: RECORD NO.=      43, CASE NO.=      11, ID NO.= 8508, CARD NO.= 3
          0850803777 7777777777 7777777000 0888888108 8811111111 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      46, CASE NO.=      12, ID NO.= 8601, CARD NO.= 2
          0860102085 1903120000 0000011111 1118555555 4544547111 1000011111 111900
0000 1000000000
          Q11 NOT SELF-CONSISTENT.
          COLUMNS 52-55
          COLUMNS 56-57

 ERROR: RECORD NO.=      51, CASE NO.=      13, ID NO.= 8602, CARD NO.= 3
          0860203433 4477347437 7777777000 0888888108 8851110111 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39

 ERROR: RECORD NO.=      55, CASE NO.=      14, ID NO.= 8603, CARD NO.= 3
          0860303747 7777747767 6774774000 0888888108 8841111810 1000000000 000000
0000 0000000000
          Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
          COLUMN 39
```

Figure 3-7 Sample Data Cleaning

Initial CLEAN Run

(continued)

```
 ERROR: RECORD NO.=      58, CASE NO.=      15, ID NO.= 8604, CARD NO.= 2
        0860402013 1807031013 0100011431 1188844443 3432413411 4000100331 411201
1010 1000000000
        Q15 NOT SELF-CONSISTENT.
        COLUMNS 65-70
        COLUMN 71

 ERROR: RECORD NO.=      59, CASE NO.=      15, ID NO.= 8604, CARD NO.= 3
        0860403424 7777116767 5475754000 0888888108 8871111110 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=      63, CASE NO.=      16, ID NO.= 8701, CARD NO.= 3
        0870103225 7276477567 7777762000 0888888108 8851101101 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=      67, CASE NO.=      17, ID NO.= 8703, CARD NO.= 3
        0870303246 6371577747 5475747000 0888888108 8851100101 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=      68, CASE NO.=      17, ID NO.= 8703, CARD NO.= 4
        0870304241 6010702120 1222505011 0000000000 0000000000 0000000000 000000
0000 0000000000
        B6 OUT OF RANGE: =0
        COLUMN 31

 ERROR: RECORD NO.=      75, CASE NO.=      19, ID NO.= 8901, CARD NO.= 3
        0890103677 7777767767 7777765000 0888888108 8851110101 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=      79, CASE NO.=      20, ID NO.= 8902, CARD NO.= 3
        0890203244 5377767777 7777747000 0888888108 8870181181 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39
```

Figure 3-7 Sample Data Cleaning

Initial CLEAN Run

(continued)

```
 ERROR: RECORD NO.=      80, CASE NO.=      20, ID NO.= 8902, CARD NO.= 4
        0890204301 4010699120 4252907012 1000000000 0000000000 0000000000 000000
0000 0000000000
        TOT.NO. ADULT RESIDENTS, COL 22,  OF CARD 1 LARGER THAN Q29, COLS 20-21,
 OF CARD 4
        IMAGE OF PREVIOUS CARD 1:
        0890201121 0890040233 6512300000 0000000000 0000000000 0000000000 000000
0000 0000000000

 ERROR: RECORD NO.=      83, CASE NO.=      21, ID NO.= 8904, CARD NO.= 3
        0890403622 7477667766 7777726000 0888888108 8821111000 8000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=      91, CASE NO.=      23, ID NO.= 9001, CARD NO.= 3
        0900103774 7777664777 7777777000 0888888108 8811111001 0000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ****** END-OF-FILE ENCOUNTERED ******
        TOTAL CARDS READ =   92

 ******     PROGRAM    ENDS      ******
@FIN
```

Figure 3-7 Sample Data Cleaning

Initial CLEAN Run

(continued)

```
@RUN,R/RT EDIT/35/0, acct-no/user-id,VINNIE,003,300
@ASG,A JACKIES.
@BRKPT PRINT$/JACKIES
@PASSWD password
@CHG,Z RAWDATA.
@ASG,A RAWDATA.
@US*ER.ED RAWDATA.
11
R,39 8
15
R,39 8
19
R,39 8
23
R,39 8
27
R,39 8
30
R,39 8
35
R,39 8
36
R,20 03
43
R,39 8
46
R,57 0
51
R,39 8
55
R,39 8
58
R,71 0
59
R,39 8
68
R,31 9
75
R,39 8
80
R,20 05
83
R,39 8
91
R,39 8
EXIT
@FREE RAWDATA.
@CHG,V RAWDATA.
@BRKPT PRINT$
@FIN
```

Figure 3-7 Sample Data Cleaning

Edit Runstream

(continued)

```
@CHG,Z RAWDATA.
FURPUR 27R3A-2     E33 SL73R1 11/08/79 10:45:45
   CHANGED!
@ASG,A RAWDATA.
@US*ER.ED RAWDATA.
ED 29B 11/08/79 10:45 (0):F
EDIT
08411034337557677277777777570000888888108881811111111
08411034337557677277777777570000888888188881811111111
08412034777737777477777777770000888888108883111181101
08412034777737777477777777770000888888188883111181101
08415036677777777677777777770000888888108885111111111
08415036677777777677777777770000888888188885111111111
08501034547377767347777777770000888888108886111111111
08501034547377767347777777770000888888188886111111111
08502037777777777777777777770000888888108887110111101
08502037777777777777777777770000888888188887110111101
085030201301111210100000011145534824117321333361111000001111211500000001000
085030201301111210100000011145534824118321333361111000001111211500000001000
08506036677667777677777777770000888888108888110111111
08506036677667777677777777770000888888188888110111111
0850604281401080312022521030121
0850604281401080312032521030121
08508037777777777777777777770000888888108881111111111
08508037777777777777777777770000888888188881111111111
086010208519031200000000011111111855555545445471111000011111111900000000100
086010208519031200000000011111111855555545445471111000010111111900000000100
08602034334477347437777777770000888888108885111101111
08602034334477347437777777770000888888188885111101111
08603037477777747767677477740000888888108884111118101
08603037477777747767677477740000888888188884111118101
086040201318070310130100011431118884444334324134114000100331411201101010100
086040201318070310130100011431118884444334324134114000100331411201101010000
08604034247777116767547575400008888881088871111111101
08604034247777116767547575400008888881888871111111101
0870304241601070212012225050 11
08703042416010702120122250501 19
08901036777777767767777776500008888881088885111101011
08901036777777767767777776500008888881888885111101011
0890204301401069912042529070121
0890204301401069912052529070121
08904036227477667766777772600008888831088882111110008
08904036227477667766777772600008888981888882111110008
09001037747777664777777777700008888881088881111110010
09001037747777664777777777700008888881888881111110010
END EDIT 92 LINES OUTPUT
@FREE RAWDATA.
@CHG,V RAWDATA.
FURPUR 27R3A-2     E33 SL73R1 11/08/79 10:45:57
   CHANGED!
@BRKPT PRINT$
```

Figure 3-7 Sample Data Cleaning
Edit Results

42<

(continued)

```
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A RAWDATA.
@USE 10,RAWDATA.
@XQT LISTEN*ATTITUDINAL.CLEAN


 *** ATTITUDINAL SURVEY DATA CLEANING BEGINS ***
 PROGRAM WRITTEN BY:  DR. RAGU SINGH, CSC
 LAST EDIT BY:  STEVEN G. EAGLESON, CSC, FALLS CHURCH, VA.
        ********VERSION 1.0(0)*********


      FLEXIBLE DATA RANGES:
      LOID=     1  HIID= 9999
      LONZ=     0  HINZ=   50
      LOAR=     0  HIAR=   15
      LOCLUS=     1  HICLUS=  400
      CITIES=   99      4     88     88
      MAX INT DATE=   60
      MAX NO OF ERRORS=    90

      WITH COMMUNITY NOISE PROGRAM
      AFFECTS Q18, Q19 Q19A, Q21


 ERROR: RECORD NO.=        63, CASE NO.=        16, ID NO.= 8701, CARD NO.= 3
        0870103225 7276477567 7777762000 0999999108 9951101101 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=        67, CASE NO.=        17, ID NO.= 8703, CARD NO.= 3
        0870303246 6371577747 5475747000 0999998108 9951100101 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ERROR: RECORD NO.=        79, CASE NO.=        20, ID NO.= 8902, CARD NO.= 3
        0890203244 5377767777 7777747000 0888888108 8870181181 1000000000 000000
0000 0000000000
        Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
        COLUMN 39

 ****** END-OF-FILE ENCOUNTERED ******
        TOTAL CARDS READ =   92

 ******      PROGRAM    ENDS      ******
@FIN
```

Figure 3-7 Sample Data Cleaning

Second CLEAN Run

(continued)

```
@RUN,R/RT EDIT/35/0,acct-no/user-id,VINNIE,003,300
@ASG,A JACKIE2.
@BRKPT PRINT$/JACKIE2
@PASSWD password
@CHG,Z RAWDATA.
@ASG,A RAWDATA.
@US*ER.ED RAWDATA.
63
R,39 8
67
R,39 8
79
R,39 8
EXIT
@FREE RAWDATA.
@CHG,V RAWDATA.
@BRKPT PRINT$
@FIN
```

```
** @PASSWD STATEMENT IGNORED **
@CHG,Z RAWDATA.
FURPUR 27R3A-2     E33 SL73R1 11/08/79 11:35:20
  CHANGED!
@ASG,A RAWDATA.
@US*ER.ED RAWDATA.
ED 29B 11/08/79 11:35 (0):F
EDIT
087010322572764775677777762000088888810888511011011
087010322572764775677777762000088888818888511011011
087030324663715777475475747000088888810888511001011
087030324663715777475475747000088888818888511001011
089020324453777677777777747000088888810888701811811
089020324453777677777777747000088888818888701811811
                                             ------
END EDIT 92 LINES OUTPUT
@FREE RAWDATA.
@CHG,V RAWDATA.
FURPUR 27R3A-2     E33 SL73R1 11/09/79 11:35:30
  CHANGED!
@BRKPT PRINT$
```

Figure 3-7 Sample Data Cleaning

Edit Runstream and Edit Run

(continued)

```
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A RAWDATA.
@USE 10,RAWDATA.
@XQT LISTEN*ATTITUDINAL.CLEAN


 *** ATTITUDINAL SURVEY DATA CLEANING BEGINS ***
 PROGRAM WRITTEN BY:  DR. RAGU SINGH, CSC
 LAST EDIT BY:  STEVEN G. EAGLESON, CSC, FALLS CHURCH, VA.
          *********VERSION 1.0(0)*********


       FLEXIBLE DATA RANGES:
       LOID=     1  HIID= 9999
       LONZ=     0  HINZ=   50
       LOAR=     0  HIAR=   15
       LOCLUS=     1  HICLUS=  400
       CITIES=   99       4      89      88
       MAX INT DATE=   60
       MAX NO OF ERRORS=   90

       WITH COMMUNITY NOISE PROGRAM
       AFFECTS Q18, Q19 Q19A, Q21


 ****** END-OF-FILE ENCOUNTERED ******
        TOTAL CARDS READ =   92

 ******      PROGRAM    ENDS       ******
@FIN
```

Figure 3-7 Sample Data Cleaning

Final CLEAN Run

(continued)

3.2.5 CONSTRAINTS/LIMITATIONS

CLEAN has two types of limitations;

- Those inherent in the type of data verification CLEAN is designed to perform
- Those created by the operating system

The first type of limitation determines the type of data errors CLEAN detects. It detects values for a variable that are either illegal for the variable to assume or are inconsistent with the values of some similar variable.

An example in this survey is question B1, sex of respondent. The legal values for this variable are 1,2,8,9. If 7 is punched on the card, it is obviously a mistake. CLEAN will notice this and warn the user.

CLEAN will also notice if the sex reported by question B1 and that reported by question A3C are different. If they are, this again is a mistake.

What CLEAN will not reveal is the case where both answers are consistent but wrong, e.g., where the number entered for both B2 and A3C is 1, meaning male, but the respondent is a woman. Nor can it determine which of two inconsistent responses is correct. The CLEAN program is no substitute for careful coding and keypunching.

Operating system limitations affect the way data is handled and therefore also affect the types of errors detectable by CLEAN. First, the operating system considers blanks to be equivalent to 0's. So, if a blank column is read by CLEAN, it proceeds exactly as if it had read a 0. This should cause few problems since, for most variables, zero is not a legal value and the error will be caught. Second, the read method used by CLEAN causes the operating system to change all the non-numeric characters to 0's. This change is accompanied by an error message, which will alert the user that an error has occurred and that the data must be changed.

3.2.6 DATA BASE

CLEAN uses one file as a data base: proj-id*RAWDATA. The file is used by CLEAN as input and is modified later by the editing runstream described in Section 3.2.3.1 to correct any errors discovered by CLEAN.

3.2.7 CONTROL LANGUAGE None.

3.2.8 INPUT

CLEAN takes input from two sources. The first is the proj-id*RAWDATA.file that is created at initialization time (see Section 3.1 and Figure 3-2). The format of the survey data in this file is shown in Table 3-1. It takes four records to describe the information resulting from one survey questionnaire. This is referred to as a 'case'.

CLEAN's second source of input is the card reader. CLEAN reads the initialization card described in Section 3.2.2 from this device.

3.2.9 OUTPUT

CLEAN produces one form of output: a list of error messages that is printed on the line printer at the site where the job has been submitted. A listing of these error messages can be found in Appendix C.

Most of CLEAN's messages are self-explanatory. A few may be difficult to interpret at first. The following paragraphs should clarify the troublesome ones:

- Q11(A-F), card 2, columns 52-57.
  Given that any of the parts A-D of this question is 1 then E and F must be zero. If E or F is 1 then A-D must be zero. E and F cannot both be 1.

- Q15(A-G), card 2, columns 65-71.
  If there is a 1 response for any part A-F then G must be zero. If G is 1 then A-F should be zero.

The treatment of the answers to questions 18 through 19A is complicated by the survey changes. If CTLPRG=1 for the CLEAN initialization (which indicates there is no noise control program in the community being dealt with) then the requirements are:

- Q18(A-F), card 3, columns 32-37.
  If there is a 1 response for A-E then F must be zero.
  If F is 1, then A-E must be zero.

- Q19, Q19A(A-D), card 3, columns 38,39-42.
  If Q19=1, then Q19A(A-D) must be 1,2,8 or 9.
  If Q19=2 then Q19A(A-D) must be zero.

47<

For CTLPRG=2, i.e., there is a noise control program, then the requirements are:

- Q18(A-F), card 3, columns 32-37.
  Q18(A-F) must all be coded as 8.
- Q19, Q19A(A-D), card 3, columns 38,39-42.
  Q19A(B-D) must all be coded 8.
  If Q19=1, then Q19AA must be 1,2,3,4,8 or 9.
  If Q19=2, then Q19AA must be 8.

The following are independent of CTLPRG:

- Q24A and Q25A, card 4, columns 12-13, 14-15.
  If Q24A is 80, then Q25A must be zero.
  If Q24A is not 80, then Q25A cannot be zero.

3.3 VARIANCE

VARIANCE calculates a number of statistics for certain attitudinal survey variables. The variables for which it calculates statistics are shown in Figure 3-8. The statistics it calculates are: mean, mean squared error, the standard error, confidence intervals and the coefficient of variation. These statistics give the user an idea as to how faithfully the survey's results reflect the feelings of the community as a whole.

3.3.1 SECURITY (See Section 3.1.1)

3.3.2 INITIATION PROCEDURES

The files proj-id*GOODDATA. and proj-id*VARIANOUTPUT., must be available. If the system has run correctly to this point, they will be.

3.3.3 OPERATION PROCEDURES

A runstream for running VARIANCE is shown in Figure 3-9. The user should substitute the values that apply to him for the parameters user-id, acc-num, proj-id, password, and site-id. The runstream should be punched onto cards and submitted from the user's remote batch station.

The MOS referred to in the figure is the community's Measure of Size. It is the total number of households in the community and is calculated before the administration of the survey (see Chapter 3, Reference 1, Appendix B). The user should punch the value for his community, starting in column 1, as a five digit number, adding any leading zeros necessary to make it five digits.

Q1
Q2
Q5A through Q5H
Q6A through Q6K
Q7
Q8
Q9
Q10A through Q10D
Q12
Q13A through Q13E
Q16A through Q16C
Q17A through Q17T
Q19
Q20
Q22
Q25A
Q28
Q29

Figure 3-8

Variables for which VARIANCE Calculates Statistics

49<

```
@RUN,R/RT VARIAN/64/0,acc-num/user-id,proj-id,005,50
@PASSWD password
@CHG,Z VARIANOUTPUT.
@ERS VARIANOUTPUT.
@ASG,A VARIANOUTPUT.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A GOODDATA.
@USE 10,GOODDATA.
@BRKPT PRINT$/VARIANOUTPUT
@XQT LISTEN*ATTITUDINAL.VARIANCE
@BRKPT PRINT$
@FREE VARIANOUTPUT.
@CHG,V VARIANOUTPUT.
@SYM,U VARIANOUTPUT.,,site-id
@FIN
```

Figure 3-9

Sample VARIANCE Runstream 50<

3.3.4 ERROR AND RECOVERY PROCEDURES (See Subsection 3.1.2)

3.3.5 CONSTRAINTS/LIMITATIONS

VARIANCE's most severe limitation is the fact that it is unable to tell good data from bad. For this reason, it is essential that the data cleaning step be performed correctly. Otherwise, the statistics produced by VARIANCE will be meaningless.

Data from up to 760 respondents can be processed by VARIANCE. This is 60 cases above the 700 recommended for the survey. This should provide plenty of extra space for zealous survey administrators.

3.3.6 DATA BASE

VARIANCE references the file proj-id*GOODDATA. It uses the data in this file to calculate its statistics. It does not alter the file.

3.3.7 CONTROL LANGUAGE None.

3.3.8 INPUT

VARIANCE has two sources of input:

- The disk file proj-id*GOODDATA.
- The runstream

Proj-id*GOODDATA. is created after a successful run of CLEAN by copying proj-id*RAWDATA. as described in 3.2.3 above. It contains the data for which VARIANCE will calculate statistics. The format and content of proj-id*RAWDATA. are described in 3.2.8.

The runstream gives VARIANCE the measure of size (MOS) for the community. It is on a single card in FORTRAN I5 format. See 3.3.3 above for information on its significance.

3.3.9 OUTPUT

VARIANCE produces two forms of output: a printer listing of the statistics produced and a disk file which contains a copy of this listing for later use. An example of this output can be seen in Figure 3-10.

```
VARIABLE          MEAN SQ.   STANDARD      CONFIDENCE      COEFFICIENT    NUMBER OF    NUMBER OF
  NAME     MEAN    ERROR       ERROR        INTERVAL      OF VARIATION    CLUSTERS    DATA POINTS
--------   ----   --------   ---------     ----------     ------------    ---------   -----------

  Q2      2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6A     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6B     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6C     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6D     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6E     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6F     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6G     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6H     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6I     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6J     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
  Q6K     2.250    .2906       .5391      3.307   1.193       .240           4.            4.
```

Figure 3-10
Sample of VARIANCE Output

3.4 TAILOR

TAILOR is included in the ASDAS to perform three functions:

- To generate the correct SPSS programs for local conditions;
- To allow the addition of acoustical monitoring data to the SPSS program RECODE2;
- To permit local additions to these programs.

These are necessary because different localities will require slightly different treatment of the attitudinal data they have collected, depending upon whether or not they have a noise control program and whether or not they have completed the acoustical survey.

All three functions are controlled by the data card that occurs directly after the @XQT ATTITUDINAL.TAILOR card in the runstream. TAILOR reads this card to determine if a noise control program exists and if acoustical monitoring data is available.

This information is coded as two numbers, or flags, punched on the card in FORTRAN 2I5 format. The first flag, in columns 1-5, refers to the availability of acoustical data. If the surveyed community has been acoustically surveyed, the flag is set to a non-zero value to indicate this. It is 0 or left blank, otherwise. The second flag, in columns 6-10, is set to a non-zero number if the community has a noise control program. If there is no such program, it is 0 or blank. The flag card must always be provided even if the community has no noise control program and has not performed the acoustical survey. An example of this card is shown on line 15 of Figures 3-11 to 3-16.

If no acoustical monitoring program exists, all that the user need do is run TAILOR on each of the SPSS elements in LISTEN*ATTITUDINAL.: RECODE1, LABELS, RECODE2, FREQUENCIES, CROSSTABS, and REGRESSIONS.

If an acoustical monitoring program does exist, it will be necessary to make use of TAILOR's second function. This function adds program statements to the SPSS program RECODE2. These statements are punched by the user according to the format given on page 4-53 of reference 1,**

**Note: As described by Figure 3-13a, the format of Reference 1 is not compatible with the SPSS version used at NCC. The RECODE of DBLEVEL should be broken up into several (3 or 4) smaller multi-line statements instead of 1 large statement.

53<

```
@RUN,R/RT TALR1/35/Ø,acc-num/user-id,proj-id,ØØ3,3Ø
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
           (leave blank card here)
ADD LISTEN*ATTITUDINAL.RECODE1
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    Ø    1
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.RECODE1
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-11

Runstream to run TAILOR on RECODE1

(Community has Noise Control But no Acoustical Monitoring)

54<

```
@RUN,R/RT TALR2/35/0,acc-num/user-id,proj-id,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
           (leave blank card here)
ADD LISTEN*ATTITUDINAL.LABELS
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    0    1
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.LABELS
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-12

Runstream to run TAILOR on LABELS

(Community has Noise Control But no Acoustical Monitoring)

```
@RUN,R/RT TALR3/35/0,acc-num/user-id,proj-id,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
           (leave blank card here)
ADD LISTEN*ATTITUDINAL.RECODE2
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    0    1
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.RECODE2
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-13

Runstream to Run TAILOR on RECODE2

(Community has Noise Control But No Acoustical Monitoring)

```
@RUN,R/RT TALR3/35/0,acc-num/user-id,proj-id,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
           (leave blank card here)
ADD LISTEN*ATTITUDINAL.RECODE2
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    1    1
COMPUTE         DBLEVEL=CL
RECODE          DBLEVEL (1=55)  (2=60)  (3=80)  (4=63) . . .
                  .
                  .
                  .
                (37=92) (38=93) (39=50) (40=73) (41=62) . . .
RECODE          DBLEVEL (46=95) (47=79) (48=61) (49=50) . . .
                  .
                  .
                  .
                (78=50) (79=66) (80=53) (81=95) (82=58) . . .
RECODE          DBLEVEL (86=35) (87=44) (88=45) (89=46) . . .
                  .
                  .
                  .
                (96=53) (97=54) (98=55) (99=56) (100=60). . .
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.RECODE2
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-13a

Runstream to run TAILOR on RECODE2
(Community has Noise Control and Acoustical Monitoring)

**NOTE: SPSS version used at NCC requires RECODES to be broken up into several small statements not 1 large multi-line statement, as described in reference 1, pg 4-53, Appendix B.

```
@RUN,R/RT TALR4/35/0,acc-num/user-id,proj-id,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
            (leave blank card here)
ADD LISTEN*ATTITUDINAL.FREQUENCIES
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    0    1
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.FREQUENCIES
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-14

Runstream to Run TAILOR on FREQUENCIES
(Community has Noise Control But no Acoustical Monitoring)

```
@RUN,R/RT TALR5/35/0,acc-num/user-id,proj-id,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
            (leave blank card here)
ADD LISTEN*ATTITUDINAL.CROSSTABS
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    0    1
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.CROSSTABS
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-15

Runstream to Run TAILOR on CROSSTABS
(Community has Noise Control But No Acoustical Monitoring)

```
@RUN,R/RT TALR6/35/0,acc-num/user-id,proj-id,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.
             (leave blank card here)
ADD LISTEN*ATTITUDINAL.REGRESSIONS
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
    0    1
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.REGRESSIONS
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-16

Runstream to Run TAILOR on REGRESSIONS

(Community has Noise Control But No Acoustical Monitoring)

Appendix B. They are placed directly after the control card described above, but only when TAILOR is run on LISTEN*ATTITUDINAL.RECODE2. The other SPSS programs are treated exactly as before.

TAILOR's third function is to add any statements to the SPSS programs that may be helpful for a particular city. The way in which they are added is very similar to the way statements are added to RECODE2. Before the user attempts to take advantage of this capability of TAILOR it is suggested that he thoroughly study the programs that will be affected by it.

3.4.1 SECURITY (See Section 3.1.1)

3.4.2 INITIATION PROCEDURES

Before TAILOR is run:

- The cards to be used as data must be punched and placed in the runstreams after the @XQT LISTEN*ATTITUDINAL.TAILOR card.
- The card with the flags on it will go first followed, in correct order, by the cards which the user wishes to insert into the program, if any.
- If there has been no acoustical survey, and the community does not wish to modify the programs, then the flag card will be the only data card following the @XQT card.
- The four files LISTEN*ATTITUDINAL., proj-id*ATTITUDINAL., proj-id*TAILORINPUT., and proj-id*TAILOROUTPUT., must all be available. The first must contain the system library and the latter three should be empty (as they will be when created by the first system initiation runstream).

3.4.3 OPERATIONAL PROCEDURES

TAILOR must be run for each of the six SPSS elements in LISTEN*ATTITUDINAL.: RECODE1 and LABELS, RECODE2, FREQUENCIES, CROSSTABS and REGRESSIONS. The runstreams shown in Figures 3-11 through 3-16 should be modified by replacing the user-id, acc-num, proj-id and password. The flag card should also be replaced with a flag card containing the appropriate values as described in Section 3.4.

61<

If acoustical survey data is to be used, the runstream in Figure 3-13a should be submitted when TAILOR is run on the RECODE2 program instead of that in Figure 3-13. Remember to include the additional data cards described in Section 3.4 directly after the card that contains the flags.

Optionally, additional data cards may be included in the runs for RECODE1, LABELS, FREQUENCIES, CROSSTABS, and REGRESSIONS in order to add statements to produce special statistics for a community. It is suggested that the user be certain he knows what he is doing before attempting this.

3.4.4 ERROR PROCEDURES

This section does not discuss errors that may be created when the user attempts to modify the SPSS programs by taking advantage of TAILOR's third function. It is presumed that a user sophisticated enough to modify the programs is also sophisticated enough to debug them.

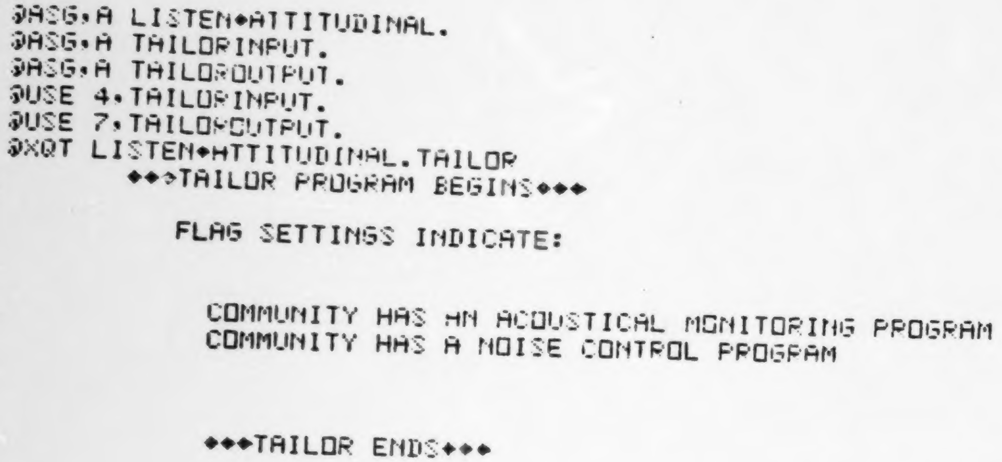
As with all the other programs, TAILOR is vulnerable to operating system and hardware failures. And, as with all the other programs, recovery from this type of failure is made by simply resubmitting the run which failed.

Unlike the other programs, TAILOR is very vulnerable to user mistakes. This is simply because there is more for the user to do in running TAILOR and therefore more opportunity for the user to make a mistake.

User mistakes can be divided into two categories: those that show up immediately, and those that remain hidden until the SPSS programs are run. Mistakes in coding the first data card, the one containing the flags, are in the first category. The user may have coded a letter instead of a number for the value of one of the flags. In this case, an error message will be printed by the operating system, the value of that flag will be set to 0 and TAILOR allowed to run to completion. The user should retype this card and resubmit the run if this happens.

The second kind of error occurs if the user codes the wrong values for the flags. To make sure that this has not happened, he should look at the output produced by each run. TAILOR prints out the values of each flag and the conditions implied by their values (see Figure 3-17). If these do not correspond to the values and conditions that should have been input, change the first data card so it contains the correct values and resubmit the run which was in error.

62<



```
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@XQT LISTEN*ATTITUDINAL.TAILOR
        **>TAILOR PROGRAM BEGINS***

             FLAG SETTINGS INDICATE:


               COMMUNITY HAS AN ACOUSTICAL MONITORING PROGRAM
               COMMUNITY HAS A NOISE CONTROL PROGRAM



               ***TAILOR ENDS***
```

Figure 3-17

Sample Output from TAILOR Program

Mistakes made in the cards following the TAILOR runstream belong in the second category. They remain hidden until the SPSS programs are run. They reveal themselves in two ways. First, the SPSS program RECODE2 may refuse to run. If so, the SPSS processor at NCC will produce an error message of the form shown in Figure 3-18. If this occurs, the user must go back and rerun TAILOR for RECODE2 after having corrected the mistake in the input cards. The mistake will probably be an obvious one such as a misspelling. Then the user should continue with the rest of the system starting with RECODE2. It is unnecessary to run RECODE1 a second time.

The second way a mistake may reveal itself is more subtle and more destructive. The program may run, but be wrong. The only way to detect this is for the user to look at the output of the RECODE2 SPSS program and check to make sure that the recodes that were performed on DBLEVEL were the ones he desired. In other words, the user should look to see that the cards used as input to TAILOR when it was run on RECODE2 are duplicated in the output of RECODE2. If they are not, the input cards for TAILOR must be corrected, the TAILOR run for RECODE2 must be resubmitted, and then the rest of the system must be continued, starting by running RECODE2 again.

This type of mistake may also be detected by looking at the output of the FREQUENCIES program. If invalid decibel levels appear, then the cards input to TAILOR when it was run on RECODE2 were wrong. This will not, however, reveal recodes that are incorrect but result in valid decibel levels.

3.4.5 DATA BASE None.

3.4.6 CONTROL LANGUAGE None.

3.4.7 INPUT

TAILOR has two sources of input: the file proj-id*TAILORINPUT. and the runstream (i.e., the logical card reader). Every time TAILOR is run, an element from LISTEN*ATTITUDINAL. is copied into proj-id*TAILORINPUT.

It is necessary to make this copy instead of operating directly on LISTEN*ATTITUDINAL. because of the way the EXEC 8 operating system

```
@RUN,R/RT TALR3/35/0,acct-no/user-id ,VINNIE,003,30
@PASSWD password
@CHG,Z ATTITUDINAL.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A TAILORINPUT.
@ASG,A TAILOROUTPUT.
@USE 4,TAILORINPUT.
@USE 7,TAILOROUTPUT.
@US*ER.ED,I TAILORINPUT.

ADD LISTEN*ATTITUDINAL.RECODE2
EXIT
@XQT LISTEN*ATTITUDINAL.TAILOR
   1     0
COMPUTE          DBLEVEL=CL
RECODE           DBLEVEL  (1=30) (2=32) (3=34) (4=36) (5=38) (6=40) (7=41) (8=43) (9=45)
                 (10=47) (11=49) (12=50) (13=51) (14=54) (15=56) (16=58) (17=60) (18=61)
                 (19=63) (20=65) (21=67) (22=69) (23=70) (24=72) (25=74) (26=76) (27=78)
                 (28=79) (29=80) (30=81) (31=8[illegible]) (32=85) (33=87) (34=89) (35=90) (36=91)
                 (37=92) (38=93) (39=94) (40=95))41=96) (42=97) (43=99) (44=100) (45=101)
RECODE           DBLEVEL  (46=103) (47=104) (48=105) (49=106) (50=107) (51=108) (52=109)
                 (53=110)
                 (54=111) (55=112) (56=113) (57=114) (58=115) (59=116) (60=117) (61=118)
                 (62=119) (63=120) (64=121) (65=122) (66=84) (67=124) (68=125) (69=126)
                 (70=127) (71=128) (72=129) (73=65) (74=45) (75=55) (76=75) (77=81)
                 (78=50) (79=66) (80=53) (81=95) (82=58) (83=81) (84=62) (85=45)
RECODE           DBLEVEL  (86=35) (87=44) (88=45) (89=46) (90=47) (91=48) (92=49)
                 (93=50)
                 (94=5) (95=52) (96=53) (97=54) (98=55) (99=56)
                 (100=60)
@EOF
@COPY,I TAILOROUTPUT.,ATTITUDINAL.RECODE2
@FREE ATTITUDINAL.
@CHG,V ATTITUDINAL.
@FIN
```

Figure 3-18

TAILOR Insertion of Acoustical Data into RECODE2

(Note circled error, results next page)

(continued)

```
                    COMMENT             SINCE WE HAVE ACOUSTICAL MEASUREMENTS, WE USE
                                        THEM TO CREATE -DBLEVEL-.
                    COMPUTE             DBLEVEL=CL
                    RECODE              DBLEVEL  (1=30) (2=32) (3=34) (4=36) (5=38) (6=40) (
7=41) (8=43) (9=45)
                                        (10=47) (11=49) (12=50) (13=51) (14=54) (15=56) (16
=58) (17=60) (18=61)
                                        (19=63) (20=65) (21=67) (22=69) (23=70) (24=72) (25
=74) (26=76) (27=78)
                                        (28=79) (29=80) (30=81) (31=83) (32=85) (33=87) (34
=89) (35=90) (36=91)
                                        (37=92) (38=93) (39=94) (40=95) )41=96) (42=97) (43
=99) (44=100) (45=101)
ERROR NUMBER..  201. PROCESSING CEASES, ERROR SCAN CONTINUES.

EPA PROTOTYPE:  VARIABLE FILE GENERATOR PART 2
    11/08/79          PAGE    5

                    RECODE              DBLEVEL  (46=103) (47=104) (48=105) (49=106) (50=1
07) (51=108) (52=109)
                                        (53=110)
                                        (54=111) (55=112) (56=113) (57=114) (58=115) (59=1
16) (60=117) (61=118)
                                        (62=119) (63=120) (64=121) (65=122) (66=84) (67=12
4) (68=125) (69=126)
                                        (70=127) (71=128) (72=129) (73=65) (74=45) (75=55)
(76=75) (77=81)
                                        (78=50) (79=66) (80=53) (81=95) (82=58) (83=81) (84
=62) (85=45)
                    RECODE              DBLEVEL  (86=35) (87=44) (88=45) (89=46) (90=47) (9
1=48) (92=49)
                                        (93=50)
                                        (94=5) (95=52) (96=53) (97=54) (98=55) (99=56)
                                        (100=60)

                    COMMENT
                                        CREATE -DBLVLSQR- FROM -DBLEVEL-

                    COMPUTE             DBLVLSQR=DBLEVEL◆DBLEVEL
```

Figure 3-18 (continued)

Attempt to Run RECODE2 with Syntax Error

66<

differentiates between program and data files. Each of the six times TAILOR is run, a different element of LISTEN*ATTITUDINAL. is copied into proj-id*TAILORINPUT. The elements to be used are: RECODE1, LABELS, RECODE2, FREQUENCIES, CROSSTABS, and REGRESSIONS.

The runstream contains the data cards that control TAILOR: the flag card and the modification cards, both described in Subsections 3.4 and 3.4.2 above. These cards follow the @XQT LISTEN*ATTITUDINAL.TAILOR card in the runstream and precede the next control card (i.e., the next card beginning with an @ sign).

3.4.8 OUTPUT

TAILOR produces two forms of output for each run it makes. The first is an element in the file proj-id*ATTITUDINAL. The elements TAILOR creates are: RECODE1, LABELS, RECODE2, FREQUENCIES, CROSSTABS, and REGRESSIONS.

The second is a printout which tells the user the values of the flags TAILOR has read in. (See Figure 3-17).

3.5 RECODE1

RECODE1 is an SPSS program which takes the verified data produced by the CLEAN program and performs a number of manipulations on it. These manipulations make the data suitable for input to the series of statistical analysis programs that follow RECODE2. The details of the manipulations are not important to the user. They may be found in the program documentation.

3.5.1 SECURITY (See subsection 3.1.1)

3.5.2 INITIATION PROCEDURES

Before running RECODE1 it is necessary to have run CLEAN and TAILOR as described in Sections 3.2 and 3.4 above. The files proj-id*GOODDATA., proj-id*SYSFILE1. and proj-id*ZSCORES. must be available.

3.5.3 OPERATIONAL PROCEDURE

To run RECODE1 the user should submit the runstream shown in Figure 3-19.

3.5.4 ERROR AND RECOVERY PROCEDURES (See subsection 3.1.2)

3.5.5 CONSTRAINTS/LIMITATIONS

RECODE1's only limitation is the number of cases of questionnaire data its output files (proj-id*SYSFILE1. and proj-id*ZSCORES.) can hold. This number is much larger than the 700 cases that the survey is designed to produce. RECODE1, like all the other SPSS programs, is unable to distinguish between good and bad data. It is therefore essential that the data cleaning step be done correctly.

3.5.6 DATA BASE

RECODE1 references two files that could be construed as data bases. The first is proj-id*GOODDATA. which contains cleaned questionnaire data. RECODE1 reads from this file. The second is proj-id*SYSFILE1. This contains the results of all the calculations done by RECODE1 as well as the data from proj-id*GOODDATA. It is in SPSS system file format. This file will be used by RECODE2.

3.5.7 CONTROL LANGUAGE None.

```
@RUN,R/RT RCOD1/64/0,acc-num/user-id,proj-id,005,50
@PASSWD password
@CHG,Z SYSFILE1.
@CHG,Z ZSCORES.
@ERS SYSFILE1.
@ERS ZSCORES.
@ASG,A LIB*SPSS7-2.
@ASG,A ATTITUDINAL.
@ASG,A SYSFILE1.
@ASG,A ZSCORES.
@ASG,A GOODDATA.
@USE 4,SYSFILE1.
@USE 20,ZSCORES.
@USE 8,GOODDATA.
@LIB*SPSS7-2.SPSS.
@ADD ATTITUDINAL.RECODE1
@EOF
@FREE SYSFILE1.
@FREE ZSCORES.
@CHG,V SYSFILE1.
@CHG,V ZSCORES.
@FIN
```

Figure 3-19

Sample RECODE1 Runstream

3.5.8 INPUT

RECODE1 uses the file proj-id*GOODDATA. as input. This file has the same format as proj-id*RAWDATA. which is described in subsection 3.2.8. It is on disk and contains the cleaned data from the attitudinal survey questionnaire.

3.5.9 OUTPUT

RECODE1 produces two output files: proj-id*SYSFILE1. and proj-id*ZSCORES. Both are stored on disk.

3.5.9.1 Proj-id*SYSFILE1

This file is a standard SPSS system file produced by the SPSS control card SAVE FILE. The format of this type file is described in reference 6 , Appendix B. The file will be used as input to the program RECODE2.

3.5.9.2 Proj-id*ZSCORES.

This file is an intermediate file containing data produced by RECODE1 which is to be used as input to RECODE2. It is produced by the SPSS control card sequence:

```
CONDESCRIPTIVE Q12,Q13A TO Q13E,Q19,Q20,Q21SUM
OPTIONS        3
```

Its format is given in reference 6 , Appendix B.

3.6 RECODE2.

RECODE2 accepts as input the files proj-id*SYSFILE1. and proj-id*ZSCORES. The data from proj-id*ZSCORES. is modified and then merged with that from proj-id*SYSFILE1. to produce the output file, proj-id*SYSFILE2. proj-id*SYSFILE2. is used by the statistical analysis programs that follow. The details of this manipulation are in the program documentation.

3.6.1 SECURITY (See subsection 3.1.1)

3.6.2 INITIATION PROCEDURES

RECODE2 can be run only after RECODE1 has executed. The data files proj-id*SYSFILE1., proj-id*ZSCORES. and proj-id*SYSFILE2. must be available for RECODE2 to run.

3.6.3 OPERATIONAL PROCEDURES

The user should submit the runstream given in Figure 3-20.

3.6.4 ERROR AND RECOVERY PROCEDURES (See subsection 3.1.2).

3.6.5 CONSTRAINTS AND LIMITATIONS (See subsection 3.5.5).

3.6.6 DATA BASE

RECODE2 reads proj-id*SYSFILE1. The information from that file is combined with that in proj-id*ZSCORES. and with the compound variables defined in RECODE2 to create a new SPSS system file: proj-id*SYSFILE2. The files proj-id*SYSFILE1. and proj-id*ZSCORES. remain on disk, unaltered.

3.6.7 CONTROL LANGUAGE None.

3.6.8 INPUT (See subsection 3.5.9).

3.6.9 OUTPUT

RECODE2 produces a disk file named proj-id*SYSFILE2. It is in SPSS system file format as described in reference 6, Appendix B. It holds the contents of proj-id*SYSFILE1. as well as a number of new, compound variables for use by later statistical analysis programs.

```
@RUN,R/RT RCOD2/64/0,acc-num/user-id,proj-id,005,40
@PASSWD password
@CHG,Z SYSFILE2.
@ERS SYSFILE2.
@ASG,A ATTITUDINAL.
@ASG,A LIB*SPSS7-2.
@ASG,A SYSFILE1.
@ASG,A ZSCORES.
@ASG,A SYSFILE2.
@USE 3,SYSFILE1.
@USE 4,SYSFILE2.
@USE 8,ZSCORES.
@LIB*SPSS7-2.SPSS
@ADD ATTITUDINAL.RECODE2
@EOF
@FREE SYSFILE2.
@CHG,V SYSFILE2.
@FIN
```

Figure 3-20

Sample RECODE2 Runstream

3.7 FREQUENCIES

FREQUENCIES produces frequency distributions and the mode, range, minimum, and maximum statistics for a number of variables in the attitudinal survey.

3.7.1 SECURITY (See subsection 3.1.1).

3.7.2 INITIATION PROCEDURES

FREQUENCIES follows RECODE2 in the ASDAS. The input file proj-id*SYSFILE2., previously generated by RECODE2, must be available before running this program. The file proj-id*FREQOUT. must also be available.

3.7.3 OPERATIONAL PROCEDURES

A runstream is given in Figure 3-21. The user should submit this runstream after substituting the appropriate values for user-id, acc-num, proj-id and password.

3.7.4 ERROR AND RECOVERY PROCEDURES (See subsection 3.1.2).

3.7.5 CONSTRAINTS/LIMITATIONS (See subsection 3.5.5).

3.7.6 DATA BASE

FREQUENCIES reads proj-id*SYSFILE2.

3.7.7 CONTROL LANGUAGE None.

3.7.8 INPUT. (See subsections 3.6.9 and 3.6.6 regarding proj-id*SYSFILE2.)

3.7.9 OUTPUT

FREQUENCIES' output file, proj-id*FREQOUT., is kept on disk and is a standard EXEC 8 alternate print file. A listing of this file is sent to the user's line printer.

73<

```
@RUN,R/RT FREQ/64/0,acc-num/user-id,proj-id,008,600
@PASSWD password
@CHG,Z FREQOUT.
@ERS FREQOUT.
@ASG,A LIB*SPSS7-2.
@ASG,A SYSFILE2.
@ASG,A ATTITUDINAL.
@ASG,A FREQOUT.
@USE 3,SYSFILE2.
@BRKPT PRINT$/FREQOUT
@LIB*SPSS7- 2.SPSS
@ADD ATTITUDINAL.FREQUENCIES
@EOF
@BRKPT PRINT$
@FREE FREQOUT.
@CHG,V FREQOUT.
@SYM,U FREQOUT.,,site-id
@FIN
```

Figure 3-21

Sample FREQUENCIES Runstream

3.8 CROSSTABS

CROSSTABS is an SPSS program which produces contingency tables with chi-square statistics for selected Attitudinal Survey variables.

3.8.1 SECURITY (See subsection 3.1.1).

3.8.2 INITIATION PROCEDURES

Before starting CROSSTABS the user must make sure that the files proj-id*SYSFILE2. and proj-id*XTABSOUT. are available. This means RECODE2 must have been run.

3.8.3 OPERATIONAL PROCEDURES

A template of the CROSSTABS runstream is shown in Figure 3-22. The user should submit this runstream after substituting for user-id, acc-num, proj-id, password, and site-id.

3.8.4 ERROR AND RECOVERY PROCEDURES (See subsection 3.1.2).

3.8.5 CONSTRAINT/LIMITATIONS (See subsection 3.5.5).

3.8.6 DATA BASE

CROSSTABS reads proj-id*SYSFILE2.

3.8.7 CONTROL LANGUAGE None.

3.8.8 INPUT (See subsections 3.6.9 and 3.6.6 regarding proj-id*SYSFILE2.)

3.8.9 OUTPUT

CROSSTABS produces the file proj-id*XTABSOUT. which is a standard EXEC 8 alternate print file. A listing of this file is sent to the user's line printer.

```
@RUN,R/RT XTABS/64/0,acc-num/user-id,proj-id,010,600
@PASSWD password
@CHG,Z XTABSOUT.
@ERS XTABSOUT.
@ASG,A LIB*SPSS7-2.
@ASG,A SYSFILE2.
@ASG,A ATTITUDINAL.
@ASG,A XTABSOUT.
@USE 3,SYSFILE2.
@BRKPT PRINT$/XTABSOUT
@LIB*SPSS7-2.SPSS
@ADD ATTITUDINAL.CROSSTABS
@EOF
@BRKPT PRINT$
@FREE XTABSOUT.
@CHG,V XTABSOUT.
@SYM,U XTABSOUT.,,site-id
@FIN
```

Figure 3-22

Sample Runstream for CROSSTABS

3.9 REGRESSIONS

REGRESSIONS is an SPSS program that performs a multiple linear regression analysis on a number of the variables contained in the Attitudinal Survey. The regressions are performed according to the causal path model outlined in reference 1, Appendix B.

3.9.1 SECURITY (See subsection 3.1.1)

3.9.2 INITIATION PROCEDURES NOT APPLICABLE.

3.9.3 OPERATIONAL PROCEDURES

See Figure 3-23 for a model REGRESSIONS runstream. Replace user-id, acc-num, proj-id, password and site-id, with the correct values and submit the runstream. Because REGRESSIONS is likely to be expensive, the runstream requests the lowest and least expensive priority. This priority means that the job probably will not run until after 5:00 P.M., NCC time.

To allow the job to run during the day, the user should punch an R instead of a D in column 6 of card 1 in the runstream (the @RUN card). The cost to run will increase about 30%.

3.9.4 ERROR AND RECOVERY PROCEDURES (See subsection 3.1.2)

3.9.5 CONSTRAINTS/LIMITATIONS (See subsection 3.5.5)

3.9.6 DATA BASE

REGRESSIONS reads proj-id*SYSFILE2.

3.9.7 CONTROL LANGUAGE None.

3.9.8 INPUT (See subsections 3.6.9 and 3.6.6)

3.9.9 OUTPUT

The output file, proj-id*REGRESSOUT., is a standard EXEC 8 alternate print file containing the results of the REGRESSION program. A listing of this file is sent to the user's line printer.

77<

```
@RUN,D/RT RGRSS/64/0,acc-num/user-id,proj-id,020,600
@PASSWD password
@CHG,Z REGRESSOUT.
@ERS REGRESSOUT.
@ASG,A LIB*SPSS7-2.
@ASG,A SYSFILE2.
@ASG,A ATTITUDINAL.
@ASG,A REGRESSOUT.
@USE 3,SYSFILE2.
@BRKPT PRINT$/REGRESSOUT
@LIB*SPSS7-2.SPSS
@ADD ATTITUDINAL.REGRESSIONS
@EOF
@BRKPT PRINT$
@FREE REGRESSOUT.
@CHG,V REGRESSOUT.
@SYM,U REGRESSOUT.,,site-id
@FIN
```

Figure 3-23

Sample REGRESSIONS Runstream

3.10 ARCHIVING THE SYSTEM'S RESULTS

The expense and trouble of running the ASDAS will have been wasted if the results are missing when they are needed at some future time. Keeping copies of the results on disk is expensive, however, and paper has a tendency to get lost. The best way to preserve the system's output is on tape. It is suggested that a separate tape be written for each city, and that is the approach taken below.

3.10.1 SECURITY

The best security that can be provided for a tape is to store it in a safe place. This place must have the right environment for tape storage and the tape must be well labeled so the data on it will not deteriorate or become unusable because its format has been forgotten. It is suggested that the tape listing portion of this section's output be used as the label. It is also suggested that a master list be maintained of tape numbers, tape names and the cities to which they correspond.

When it is written, the system archive tape will be catalogued as a public file with a write key. This means that any user can read the tape, but he must know the proper key (a sort of password) to write on it. This key will be the same for all ASDAS tapes.

3.10.2 INITIATION PROCEDURES

The user's first step in storing the ASDAS data is, of course, to run the system. Next, a tape must be obtained. To request a tape, the user should call the Tape Librarian (See Appendix A for a list of useful NCC phone numbers).

If the tape will remain at NCC, a library tape should be asked for. If it is to be removed, a 'B-tape' is needed. Users are charged tape rental and storage costs. Library tapes are much cheaper. The Tape Librarian will know the current charges.

3.10.3 OPERATIONAL PROCEDURES

Figure 3-24 shows the runstream that should be used to archive the ASDAS. As usual, the user must substitute the appropriate values for user-id, acc-num, proj-id and password. New for this run is 'tape-num' the number assigned to the tape by the Tape Librarian. It will be given to the user at the time the tape is requested.

```
@RUN,R/RT ARCHIV/35/1,acc-num/user-id,proj-id,005
@PASSWD password
@DELETE,C tape-nam.
@ASG,UP/W tape-nam.//ASDAS,36N,tape-num
@USE OUTTAPE.,tape-nam.
@ASG,A LISTEN*ATTITUDINAL.
@ASG,A ATTITUDINAL.
@ASG,A GOODDATA.
@ASG,A SYSFILE2.
@ASG,A VARIANOUTPUT.
@ASG,A FREQOUT.
@ASG,A XTABSOUT.
@ASG,A REGRESSOUT.
@COPOUT LISTEN*ATTITUDINAL.,OUTTAPE.
@COPOUT ATTITUDINAL.,OUTTAPE.
@COPY VARIANOUTPUT.,OUTTAPE.
@COPY FREQOUT.,OUTTAPE.
@COPY XTABSOUT.,OUTTAPE.
@COPY REGRESSOUT.,OUTTAPE.
@COPY GOODDATA.,OUTTAPE.
@COPY SYSFILE2.,OUTTAPE.
@REWIND OUTTAPE.
@US*ER.FLIST,B OUTTAPE.
@FIN
```

Figure 3-24

Runstream to Archive ASDAS Results

The user must also substitute some meaningful name for 'tape-nam'. This name must be twelve letters or shorter. It is suggested that the name take the form: 'city-ASDAS' where 'city' is the city's name or a shortened form of it and the dash is included in the name.

If the user wishes to write more than one archive tape, the archiving run should be submitted once per tape. Tape-nam and tape-num should be altered appropriately. It is suggested that they be named consecutively: 'city-ASDAS1', 'city-ASDAS2', etc.

The runstream saves the files:

LISTEN*ATTITUDINAL.	(Which contains CLEAN, TAILOR, VARIANCE and the raw form of all SPSS programs)
proj-id*ATTITUDINAL	(Which contains the SPSS programs in the form in which they were used on the data).
proj-id*VARIANOUTPUT.	
proj-id*FREQOUT.	
proj-id*XTABSOUT.	
proj-id*REGRESSOUT.	
proj-id*GOODDATA.	
proj-id*SYSFILE2.	

The program files are saved because there is plenty of room on the tape and because it may be very useful to know exactly which version of a program produced the output in hand. The best way to ensure that this information is available is to keep a copy of the program along with the output it has generated.

3.10.4 ERROR AND RECOVERY PROCEDURES

If for some reason, this job should fail in the middle of a run, simply resubmit it. It should run to completion. If the job consistently fails, take the input deck and the output listings to a systems programmer.

It is suggested that the runstream discussed in the next section be run to make sure that the tape has been written correctly. If it has not, check the cards used in this section. If they are incorrect, correct them and resubmit the run. If they are correct, consult a systems programmer.

3.10.5 CONSTRAINTS/LIMITATIONS Not Applicable.

3.10.6 DATA BASE None.

3.10.7 CONTROL LANGUAGE None.

3.10.8 INPUT

The inputs to the FURPUR processor (an EXEC 8 utility program that copies the files to tapes) are

```
LISTEN*ATTITUDINAL.
proj-id*ATTITUDINAL.
proj-id*VARIANOUTPUT.
proj-id*FREQOUT.
proj-id*XTABSOUT.
proj-id*REGRESSOUT.
proj-id*GOODDATA.
proj-id*SYSFILE2.
```

These are disk files. Their formats are given in the previous portions of the manual.

3.10.9 OUTPUT

The output of the run is a tape with the input files copied onto it in the order given above. The two program files, LISTEN*ATTITUDINAL. and proj*ATTITUDINAL, are in COPOUT format. The data files are in COPY format.

A directory of the files copied to the tape is also produced. It is suggested that a copy of this be kept with the tape as noted above.

3.11 DE-ARCHIVING THE SYSTEM'S RESULTS

When the information on the archive tape is needed, it will be necessary to run a tape reading job to obtain it. Such a job is also a good check on the archiving procedure of 3.10, to see that the data have been written to the tape correctly.

3.11.1 SECURITY (See section 3.10.1)

3.11.2 INITIATION PROCEDURE

A tape of the form described in Section 3.10 must be available. If the tape was removed from the computer center, it must be returned.

The user must know the tape's number and the name with which it was created. Otherwise, he will be unable to get the information off it. Check the documentation that the tape's creator should have affixed to the tape for the tape name and tape number.

3.11.3 OPERATIONAL PROCEDURE

Figure 3-25 shows the runstream that is used to read the data from the tape. User-id, acc-num, proj-id, password, and site-id are substituted for as usual. Tape-nam and tape-num should be taken from the documentation accompanying the tape to be read. If there is none, a systems programmer will have to be consulted to read this data from the label written on the tape itself.

The runstream will read the tape and copy the files it contains onto disk. The names of the de-archived disk files will be different from the names of the files that were copied to the tape. The new names are given in 3.11.9 below. Using different names prevents accidental destruction of the original files.

3.11.4 ERROR AND RECOVERY PROCEDURES

A number of errors can occur when running this job. First, the tape name or number can be incorrectly identified or incorrectly punched. If this happens, the user will probably get an error message of the form:

```
'I/O ERROR 20.'
```

In some instances, other error messages such as

```
'tape-nam IS BADLY POSITIONED OR HAS BAD LABEL'
```

may be generated. If one of these messages occurs, the user should check to see that the tape name and number are punched correctly in the runstream.

83<

```
@RUN,R/RT  DRCHIV/35/1,acc-num/user-id,proj-id,006,500
@PASSWD password
@DELETE,C RLISTNATT.
@DELETE,C RPROJATT.
@DELETE,C RVAROUT.
@DELETE,C RFREQOUT.
@DELETE,C RXTABSOUT.
@DELETE,C RREGOUT.
@DELETE,C RGDDATA.
@DELETE,C RSYS2.
@CAT RLISTNATT.
@CAT RPROJATT.
@CAT RVAROUT.
@CAT RFREQOUT.
@CAT RXTABSOUT.
@CAT RREGOUT.
@CAT RGDDATA.
@CAT RSYS2.
@DELETE,C tape-nam.
@ASG,U tape-nam.,36N,tape-num
@USE INTAPE.,tape-nam.
@REWIND INTAPE.
@COPIN INTAPE.,RLISTNATT.
@COPIN INTAPE.,RPROJATT.
@COPY INTAPE.,RVAROUT.
@COPY INTAPE.,RFREQOUT.
@COPY INTAPE.,RXTABSOUT.
@COPY INTAPE.RREGOUT
@COPY INTAPE.RGDDATA
@COPY INTAPE.RSYS2
@FREE INTAPE.
@HDG *** ELEMENTS IN PROJ-ID*RLISTNATT.***
@PRT,T RLISTNATT.
@HDG ***ELEMENTS IN PROJ-ID*RPROJATT.***
@PRT,T RPROJATT.
@HDG,N
@SYM,U RVAROUT.,,site-id
@SYM,U RFREQOUT.,,site-id
@SYM,U RXTABSOUT.,,site-id
@SYM,U RREGOUT.,,site-id
@SYM,U RGDDATA.,,site-id
@FIN
```

Figure 3-25

Runstream to De-Archive ASDAS Results

Second, the computer operator may be unable to find the tape. If so, he will send a message to that effect to the user's job and abort it. If the user is certain the tape is at NCC, the Tape Librarian should be called, the tape located and the runstream submitted again.

The final possibility is that the data on the tape may be bad. This will probably cause the run to abort with an 'I/O ERROR'. If it does not, the file listings will make it obvious that an error has occurred by being filled with garbage. There is not a great deal the user can do in this case, except hope that another copy of the tape is available.

3.11.5 CONSTRAINTS/LIMITATIONS Not Applicable.

3.11.6 DATA BASE None.

3.11.7 CONTROL LANGUAGE None.

3.11.8 INPUT

Input to this section's runstream is from a tape written by the runstream described in 3.10 above. This tape contains the ASDAS programs generated for a particular city and the output of those programs. See 3.10 for its format.

3.11.9 OUTPUT

Output from this runstream is in two forms: Files on disk and a listing of these files. The names of the files on disk are <u>not</u> the names under which they were copied to tape. This prevents the user from accidentally destroying the original data files.

The new file names are:

```
proj-id*RLISTNATT.
proj-id*RPROJATT.
proj-id*RVAROUT.
proj-id*RFREQOUT.
proj-id*RXTABSOUT.
proj-id*RREGOUT.
proj-id*RGDDATA.
proj-id*RSYS2.
```

where the initial R signifies "restored."

85<

A directory of the elements contained in the files proj-id*RLISTNATT. and proj-id*RPROJATT. is included in the runstream printout. The directory for proj-id*RPROJATT. should show that it contains the elements:

RECODE1
LABELS
RECODE2
FREQUENCIES
CROSSTABS
REGRESSIONS

Proj-id*LISTNATT. should contain these elements plus others.

Five other printouts are also produced. These printouts each list the contents of one of the following files:

proj-id*RVAROUT.
proj-id*RFREQOUT.
proj-id*RXTABSOUT.
proj-id*RREGOUT.
proj-id*RGDDATA.

3.12 CLEAN-UP

Once all work with the ASDAS has been completed, it may be desired to delete the files it uses from disk. This will save money and make those filenames available for other uses.

3.12.1 SECURITY Not applicable.

3.12.2 INITIATION PROCEDURES Not applicable.

3.12.3 OPERATIONAL PROCEDURES

Submit the runstream in Figure 3-26 after changing user-id, acc-num, proj-id and password to the correct values. The runstream deletes the following files:

proj-id*ATTITUDINAL.
proj-id*VARIANOUTPUT.
proj-id*FREQOUT.
proj-id*XTABSOUT.
proj-id*REGRESSOUT.
proj-id*GOODDATA.
proj-id*RAWDATA.
proj-id*SYSFILE1.
proj-id*ZSCORES.
proj-id*SYSFILE2.
proj-id*RLISTNATT.
proj-id*RPROJATT.
proj-id*RVAROUT.
proj-id*RFREQOUT.
proj-id*RXTABSOUT.
proj-id*RREGOUT.
proj-id*RGDDATA.
proj-id*RSYS2.

3.12.4 ERROR AND RECOVERY PROCEDURES

If this run is submitted before the user is truly done with the system, the system will have to be run again from the beginning or restored from

```
@RUN,R/RT KILL/35/0,acc-num/user-id,proj-id,S030,20
@PASSWD password
@DELETE,C ATTITUDINAL.
@DELETE,C VARIANOUTPUT.
@DELETE,C FREQOUT.
@DELETE,C XTABSOUT.
@DELETE,C REGRESSOUT.
@DELETE,C GOODDATA.
@DELETE,C RAWDATA.
@DELETE,C SYSFILE1.
@DELETE,C ZSCORES.
@DELETE,C SYSFILE2.
@DELETE,C RLISTNATT.
@DELETE,C RPROJATT.
@DELETE,C RVAROUT.
@DELETE,C RFREQOUT.
@DELETE,C RXTABSOUT.
@DELETE,C RREGOUT.
@DELETE,C RGDDATA.
@DELETE,C RSYS2.
@FIN
```

3-26 Sample Runstream to Delete LISTEN System

88<

the archive tape (if available). The only errors that will interfere with this runstream's successful execution are transitory computer failures. Deal with these by resubmitting the run.

3.12.5 CONSTRAINTS/LIMITATIONS Not applicable.

3.12.6 DATA BASE None.

3.12.7 CONTROL LANGUAGE None.

3.12.8 INPUT Not applicable.

3.12.9 OUTPUT Not applicable.

APPENDIX A

INFORMATION REGARDING THE EPA'S NATIONAL COMPUTER CENTER (NCC)

A.1 USEFUL PHONE NUMBERS

Figures A-1 through A-3 give phone numbers of interest to NCC users.

A.2 USING TAPES

To obtain tapes for use on the NCC computer, contact the tape librarian at FTS 629-2385 (Commercial 919-541-2385) and provide the following information:

a. User name
b. User account number
c. EPA Division with which user is working
d. Number of tapes required

Tape Costs:

Rental of system library tape	$2.00/month
Storage of 'B-tape' (external tape)	$2.00 (1st day)
	$1.00 (daily, days 2-14)
	$2.00 (daily, days 14-)

This does not include the purchase price of the 'B-tape'.

A.3 RUN PRIORITY

Batch runs submitted to the NCC must request one of four possible priority:

Urgent
Preferred
Routine
Deferred

Urgent requires special arrangements and permission and is almost never used. Anticipated turnaround times for the other priorities are:

90th Percentile of Turnaround Time (Minutes)

Estimated Runtime in Minutes	Preferred	Routine	Deferred
2	50	60	ON (overnight)
10	120	180	ON
30	360	ON (over night)	ON
60	ON (overnight)	ON	ON

Using the cost of a routine job as a base, the costs of an identical job run at another priority would be in these proportions:

Urgent	6.0
Preferred	2.0
Routine	1.0
Deferred	0.75

Any job requesting these resources or greater will be held till after 1700 NCC time:

66 kWords of main memory

60 minutes of execution time

30 minutes of execution time and 45 kWords of main memory

4 tape drives

The charge will not be reduced.

Runs submitted and completed between 0800 and 1700 on Saturdays receive a 50% discount.

A.4 SYSTEM AVAILABILITY

Table A-1 shows the NCC operations schedule. Note that communications are now available starting at 0730 Monday through Friday. Saturday's schedule is unchanged.

NCC SERVICES	
Account Registration	(919) 541-3629/3641 (FTS) 629-3629/3641
Assistance/Consultation	(919) 541-3649 (FTS) 629-3649
System Status Recording	(919) 541-2226/2227 (FTS) 629-2226/2227
Documentation	(919) 541-3646/3647 (FTS) 629-3646/3647
Production Control	(919) 541-3655 (FTS) 629-3655
Programming/Analysis	(919) 541-3619 (FTS) 629-3619
Refunds	(919) 541-3649 (FTS) 629-3649
Tape Librarian	(919) 541-2385 (FTS) 629-2385
Telecommunications	
NCC Technician	(919) 541-4506 (FTS) 629-4506
COMNET	(WATS) (800) 424-8647 (FTS) (202) 537-2610
Training	(919) 541-3648/3647 (FTS) 629-3648/3647
User Communications	(919) 541-3647/3646 (FTS) 629-3647/3646

Figure A-1

Useful NCC Phone Numbers

LOCATION	BAUD RATE	MODEM	TELEPHONE
Washington, D. C.	2000 baud	201-A	(202) 686-5019
	4800 baud	208-B	(202) 363-4713
All other areas	2000 baud	201-A	(800) 424-3679
	4800 baud	208-B	(800) 424-3672

Figure A-2

Phone Numbers for NCC Remote Batch Entry Access

INTERACTIVE ACCESS TO NCC

To gain access to the NCC time sharing system the user must be able to answer two questions:

1. Which speed am I running (300 bps or 1200 bps)?
2. Where am I?

For 300 bps:

Am I in a network city? —NO→ Dial: (800) 424-3690 Then type: NCCDEMD

↓ YES

Dial local number for your city
Then type: NCCDEMD

NETWORK CITIES

Athens, Ga.	(404) 549-3882	Montgomery, Ala.	(205) 277-939
Atlanta, Ga.	(404) 873-6431	Nashville, Tenn.	(615) 244-802
Boston, Mass.	(617) 742-0420	New Orleans	(504) 566-004
Chicago, Ill.	(312) 663-1640	New York, N.Y.	(212) 962-794
Cincinnati, Ohio	(513) 751-5800	Philadelphia, Pa.	(215) 925-440
Columbia, S.C.	(803) 256-1018	Res. Tri. Park, N.C.	(919) 541-200
Dallas, Tex.	(214) 651-1723	San Francisco, Cal.	(415) 546-139
Denver, Colo.	(303) 837-0843	Seattle, Wash.	(206) 682-645
Grosse Ile, Mich.	(313) 675-8936	Washington, D. C.	(202) 966-951
Kansas City, Mo.	(816) 474-3540	Wethersfield, Conn.	(203) 529-337
Las Vegas, Nev.	(702) 736-1988		

For 1200 bps:
Either VADIC 3400 series compatibility or BELL 212 compatibility

Am I in Washington, D. C.? —YES→ Dial: (202) 966-9820 Then type: NCCMSPD

↓ NO

Am I in Res. Tri. Park, N.C.? —YES→ Dial: (919) 541-2071 Then type: NCCMSPD

↓ NO

Dial (800) 424-3607
Then type: NCCDEMD

Figure A-3
Numbers to Call to Gain Access to NCC Timesharing

TABLE A-1

NCC OPERATIONS SCHEDULE

DAY OF WEEK	FULL SYSTEM PROD.	SPLIT SYSTEM PROD.	FULL SYSTEM P.M.	SPLIT SYSTEM P.M./CM	FULL SYSTEM S.A. TIME	SPLIT SYSTEM S.A. TIME	AS NEEDED FOR BACK-LOG PROC.	COMMUNI-CATIONS
MON	0600-2400		0000-0600					0800-2400
TUES	0000-0100 0800-2400	0100-0800		0300-0500		0100-0300 0500-0700		0800-2400
WED	0000-0100 0800-2400	0100-0500			0500-0700 (INSTALL NEW BOOT TAPE)	0100-0500		0800-2400
THURS	0000-0100 0800-2400	0100-0800		0300-0500		0100-0300 0500-0700		0800-2400
FRI	0000-0100 0800-2400	0100-0800				0100-0700		0800-2400
SAT	0000-1630						1630-2400	0800-1630
SUN					1400-2400		0000-1400	
TOTAL	102.5 HRS.	25 HRS.	6 HRS.	4 HRS.	12 HRS.	18 HRS.	21.5 HRS.	88.5 HRS.

P.M. - Preventive Maintenance

C.M. - Communications Maintenance

S.A. - Systems Analysis

95<

Table A-2 NCC Remote Site ID's

Region 1	Boston	DATA0B
Region 2	New York	DATA01
Region 3	Philadelphia	DATA0H
Region 4	Atlanta	DATA0C
Region 5	Chicago	DATA0D
Region 6	Dallas	DATA0E
Region 7	Kansas City	DATA0F
Region 8	Denver	DATA0M
Region 9	San Francisco	DATA0Z
Region 10	Seattle	DATA0U
NERC	Cincinnatti	DATA0Q and DATA0X
NERC	Las Vegas	DATA0S
Headquarters	Washington, D.C.	S07PR1
NCC	RTP	PR
ODW	Headquarters	DATA1C

APPENDIX B - PROJECT REFERENCES

MANUALS FOR THE ATTITUDINAL SURVEY

1. Wyle Research and the Institute for Social Science Research, Community Noise Assessment Manual Social Survey Workbook (Wyle Research Report WR77-4). El Segundo, CA: Wyle Research, 1977

2. Wyle Research and the Institute for Social Science Research, Community Noise Assessment Manual Appendices for the Social Survey Workbook. (Wyle Research Report WR77-4). El Segundo, CA: Wyle Research 1977.

3. Michael Glazer, Steve Eagleson, and Kim Nguyen, Attitudinal Survey Data Analysis System (ASDAS) Programmer's Maintenance Manual. Falls Church, VA: Computer Sciences Corporation, 1979

MANUALS FOR OTHER PORTIONS OF THE QUIET COMMUNITIES PROGRAM

4. Patrick K. Glenn, Community Noise Assessment Manual Acoustical Survey Computerized Data Reduction Procedures. (Wyle Research Report WR79-8). El Segundo, CA: Wyle Research, 1979

5. Richard E. Burke, Community Noise Strategy Guidelines Manual. (Wyle Research Report WR78-1). El Segundo, CA: Wyle Research, 1978

6. Norman H. Nie, et al, Statistical Package for the Social Sciences, Second Edition. New York, New York: McGraw-Hill, Inc., 1975

7. Norman H. Nie and C. Hadlai Hull, SPSS Batch Release 7.0 Update Manual March, 1977

REFERENCES FOR THE UNIVAC 1110, EXEC 8 AND SYSTEM PROCESSORS

8. SDC Integrated Services, NCC User Reference Manual, (10 volumes). Research Triangle Park, NC: 1977 Revised June, 1979

9. Sperry Univac Computer Systems, Sperry Univac 1100 Series Executive System Programmer Reference, (4 volumes). (Sperry documents UP-4144.11, UP-4144.21, UP-4144.31, UP-4144.41).

10. Sperry Univac Computer Systems, EXEC 8 Hardware/Software Summary. (UP-7824R1)

11. Sperry Univac Federal Systems Operations, Introduction to EXEC 8. (SST-8001). McLean, VA 1973

12. Sperry Univac Education Media Development, 1100 Executive Control Language. (UE-738R1). Princeton, NJ: Sperry Univac, 1976

13. Sperry Univac Computer Systems, 1100 Series Fortran V Programmer Reference (UP-4060R2 with updates R2-E and R2-F)

14. Sperry Univac Computer Systems, 1100 Series Fortran V Library Programmer Reference (UP-7876 updates, A, B, C, and D)

UNIVERSITY OF MARYLAND TEXT EDITOR (US*ER.ED)

15. P.E. Hagerty and K.E. Sibbald, Text Editor User's Guide (University of Maryland Computer Note CN-7.1). College Park, MD: University of Maryland Computer Science Center, 1977

STATISTICS

16. Leslie Kish, SURVEY SAMPLING.
New York, NY: John Wiley and Sons, Inc.

APPENDIX C

ERRORS REPORTED BY CLEAN

```
  ATTITUDINAL SURVEY DATA CLEANING MESSAGES                                                  DATE 092679        PAGE    1
@ASG,A DATA.
@USE 10,DATA.
@XQT CLEAN,CLEAN

 *** ATTITUDINAL SURVEY DATA CLEANING BEGINS ***
 PROGRAM WRITTEN BY:  DR. RAGU SINGH, CSC
 LAST EDIT BY:  STEVEN G. EAGLESON, CSC, FALLS CHURCH, VA.
         ********VERSION 2.4(9)*********

        FLEXIBLE DATA RANGES:
        LOID=    1  HIID= 1500
        LONZ=    1  HINZ=   17
        LOAR=    1  HIAR=    7
        LOCLUS=    1  HICLUS=  100
        CITIES=   88      88      88      88
        MAX INT DATE=   60
        MAX NO OF ERRORS=     9

        WITH COMMUNITY NOISE PROGRAM
        AFFECTS Q18, Q19 Q19A, Q21


 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        DISALLOWED COLUMNS (27-80) HAVE NUMBERS.
        COLUMN NUMBER 27 EQUALS 9
        COLUMN NUMBER 28 EQUALS 9

 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        NOISE ZONE OUT OF RANGE: = 0
        COLUMNS  8- 9

 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        AREA OUT OF RANGE: = 0
        COLUMN 10

 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        CLUSTER OUT OF RANGE: =  0
        COLUMNS 11-13

 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        CITY OUT OF RANGE: =  0
        COLUMNS 14-16

 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        SELECTION LABEL OUT OF RANGE: =55
        COLUMNS 17-18

 ERROR: RECORD NO.=          1, CASE NO.=         1, ID NO.=     1, CARD NO.= 1
```

```
ATTITUDINAL SURVEY DATA CLEANING MESSAGES                                                    DATE 092679        PAGE    2
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        NUMBER OF CALLS OUT OF RANGE: =0
        COLUMN 19

ERROR: RECORD NO.=        1, CASE NO.=        1, ID NO.=    1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        INTERVIEW DATE OUT OF RANGE: = 0
        COLUMNS 20-21

ERROR: RECORD NO.=        1, CASE NO.=        1, ID NO.=    1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        NUMBER OF ADULT RESIDENTS OUT OF RANGE: =0
        COLUMN 22

ERROR: RECORD NO.=        1, CASE NO.=        1, ID NO.=    1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        RESPONDENT NUMBER OUT OF RANGE: =9
        COLUMN 23

ERROR: RECORD NO.=        1, CASE NO.=        1, ID NO.=    1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        RESIDENT TCT=0 RESPONDENT NUM=9 NOT CONSISTENT,
        COLUMN 22, COLUMN 23 RESPECTIVELY.

ERROR: RECORD NO.=        1, CASE NO.=        1, ID NO.=    1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        SEX IS NEITHER 1 NOR 2.
        COLUMN 24

ERROR: RECORD NO.=        1, CASE NO.=        1, ID NO.=    1, CARD NO.= 1
        0000101000 0000005500 0090009900 0000000000 0000000000 0000000000 0000000000 0000000000
        AGE LESS THAN 18: = 0
        COLUMNS 25-26

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
        0000102000 5555555555 5555500400 0000000000 0000000000 0555555500 0000555555 5555000000
        Q1 OUT OF RANGE: = 0
        COLUMNS  8- 9

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
        0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q2 OUT OF RANGE: =0
        COLUMN 10

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
        0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q3A  OUT OF RANGE: =55
        COLUMNS 11-12

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
        0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q3B  OUT OF RANGE: =55
        COLUMNS 13-14

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
        0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q3C  OUT OF RANGE: =55
```

```
ATTITUDINAL SURVEY DATA CLEANING MESSAGES                                              DATE 092679        PAGE     3
       COLUMNS 15-16

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q4A  OUT OF RANGE: =555
       COLUMNS 17-19

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q4B  OUT OF RANGE: =555
       COLUMNS 20-22

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q4C  OUT OF RANGE: =555
       COLUMNS 23-25

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5A  OUT OF RANGE: =0
       COLUMN 26

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5B  OUT OF RANGE: =0
       COLUMN 27

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5C  OUT OF RANGE: =0
       COLUMN 28

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5D  OUT OF RANGE: =0
       COLUMN 29

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5E  OUT OF RANGE: =0
       COLUMN 30

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5F  OUT OF RANGE: =0
       COLUMN 31

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5G  OUT OF RANGE: =0
       COLUMN 32

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q5H  OUT OF RANGE: =0
       COLUMN 33
```

```
ATTITUDINAL SURVEY DATA CLEANING MESSAGES                                                DATE 092679          PAGE    4
ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6A  OUT OF RANGE: =0
       COLUMN 34

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6B  OUT OF RANGE: =0
       COLUMN 35

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6C  OUT OF RANGE: =0
       COLUMN 36

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6D  OUT OF RANGE: =0
       COLUMN 37

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6E  OUT OF RANGE: =0
       COLUMN 38

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6F  OUT OF RANGE: =0
       COLUMN 39

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6G  OUT OF RANGE: =0
       COLUMN 40

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6H  OUT OF RANGE: =0
       COLUMN 41

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6I  OUT OF RANGE: =0
       COLUMN 42

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6J  OUT OF RANGE: =0
       COLUMN 43

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q6K  OUT OF RANGE: =0
       COLUMN 44

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
```

```
        Q7 OUT OF RANGE: =0
        COLUMN 45

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q8 OUT OF RANGE: =0
        COLUMN 46

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q9 OUT OF RANGE: =0
        COLUMN 47

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q10A  OUT OF RANGE: =0
        COLUMN 48

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q10B  OUT OF RANGE: =0
        COLUMN 49

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q10C  OUT OF RANGE: =0
        COLUMN 50

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q10D  OUT OF RANGE: =0
        COLUMN 51

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q11A  OUT OF RANGE: =5
        COLUMN 52

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q11B  OUT OF RANGE: =5
        COLUMN 53

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q11C  OUT OF RANGE: =5
        COLUMN 54

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q11D  OUT OF RANGE: =5
        COLUMN 55

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
        Q11E  OUT OF RANGE: =5
        COLUMN 56
```

```
ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q11F  OUT OF RANGE: =5
       COLUMN 57

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q11 NOT SELF-CONSISTENT.
       COLUMNS 52-55
       COLUMNS 56-57

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q12 OUT OF RANGE: =5
       COLUMN 58

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q13A  OUT OF RANGE: =0
       COLUMN 59

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q13B  OUT OF RANGE: =0
       COLUMN 60

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q13C  OUT OF RANGE: =0
       COLUMN 61

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q13D  OUT OF RANGE: =0
       COLUMN 62

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q13E  OUT OF RANGE: =0
       COLUMN 63

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q14 OUT OF RANGE: =0
       COLUMN 64

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15A  OUT OF RANGE: =5
       COLUMN 65

ERROR: RECORD NO.=        2, CASE NO.=        1, ID NO.=     1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15B  OUT OF RANGE: =5
       COLUMN 66
```

```
ATTITUDINAL SURVEY DATA CLEANING MESSAGES                                                          DATE 092679        PAGE     7
ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15C  OUT OF RANGE: =5
       COLUMN 67

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15D  OUT OF RANGE: =5
       COLUMN 68

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15F  OUT OF RANGE: =5
       COLUMN 69

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15F  OUT OF RANGE: =5
       COLUMN 70

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15G  OUT OF RANGE: =5
       COLUMN 71

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q15 NOT SELF-CONSISTENT,
       COLUMNS 65-70
       COLUMN 71

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q16A  OUT OF RANGE: =5
       COLUMN 72

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q16B  OUT OF RANGE: =5
       COLUMN 73

ERROR: RECORD NO.=        2, CASE NO.=          1, ID NO.=    1, CARD NO.= 2
       0000102000 5555555555 5555500000 0000000000 0000000000 0555555500 0000555555 5555000000
       Q16C  OUT OF RANGE: =5
       COLUMN 74

ERROR: RECORD NO.=        3, CASE NO.=          1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q17A  OUT OF RANGE: =0
       COLUMN  8

ERROR: RECORD NO.=        3, CASE NO.=          1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q17B  OUT OF RANGE: =0
       COLUMN  9

ERROR: RECORD NO.=        3, CASE NO.=          1, ID NO.=    1, CARD NO.= 3
```

```
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17C  OUT OF RANGE: =0
        COLUMN 10

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17D  OUT OF RANGE: =0
        COLUMN 11

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17E  OUT OF RANGE: =0
        COLUMN 12

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17F  OUT OF RANGE: =0
        COLUMN 13

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17G  OUT OF RANGE: =0
        COLUMN 14

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17H  OUT OF RANGE: =0
        COLUMN 15

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000001000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17I  OUT OF RANGE: =0
        COLUMN 16

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17J  OUT OF RANGE: =0
        COLUMN 17

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17K  OUT OF RANGE: =0
        COLUMN 18

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17L  OUT OF RANGE: =0
        COLUMN 19

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17M  OUT OF RANGE: =0
        COLUMN 20

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q17N  OUT OF RANGE: =0
```

```
     COLUMN 21

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q17O  OUT OF RANGE: =0
     COLUMN 22

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q17P  OUT OF RANGE: =0
     COLUMN 23

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q17Q  OUT OF RANGE: =0
     COLUMN 24

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q17R  OUT OF RANGE: =0
     COLUMN 25

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q17S  OUT OF RANGE: =0
     COLUMN 26

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q17T  OUT OF RANGE: =0
     COLUMN 27

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q18A  OUT OF RANGE: =5
     COLUMN 32

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q18B  OUT OF RANGE: =5
     COLUMN 33

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q18C  OUT OF RANGE: =5
     COLUMN 34

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q18D  OUT OF RANGE: =5
     COLUMN 35

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
     0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
     Q18E  OUT OF RANGE: =5
     COLUMN 36
```

```
ATTITUDINAL SURVEY DATA CLEANING MESSAGES
                                                                                                            DATE 092679      PAGE   10
ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q18F  OUT OF RANGE: =3
       COLUMN 37

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q18 NOT SELF-CONSISTENT.
       COLUMNS 32-36
       COLUMN 37

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
       COLUMN 39

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q19A(B-D)  NOT 8 FOR COMMUNITY WITH NOISE PROGRAM   Q19A(A-D)=0  5  5  5
       COLUMNS 39-42

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q20 OUT OF RANGE: =0
       COLUMN 43

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q21A  OUT OF RANGE: =5
       COLUMN 44

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q21B  OUT OF RANGE: =5
       COLUMN 45

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q21C  OUT OF RANGE: =5
       COLUMN 46

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q21D  OUT OF RANGE: =5
       COLUMN 47

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q21E  OUT OF RANGE: =5
       COLUMN 48

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
       0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
       Q21F  OUT OF RANGE: =5
       COLUMN 49

ERROR: RECORD NO.=        3, CASE NO.=        1, ID NO.=    1, CARD NO.= 3
```

```
ATTITUDINAL SURVEY DATA CLEANING MESSAGES                                                                      DATE 092679       PAGE   11
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q21G  OUT OF RANGE: =5
        COLUMN 50

ERROR: RECORD NO.=        3, CASE NO.=         1, ID NO.=    1, CARD NO.= 3
        0000103000 0000000000 0000000000 0555553805 5505555555 7000000000 0000000000 0000000000
        Q21H  OUT OF RANGE: =7
        COLUMN 51

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q23 NOT CONSISTENT WITH Q22: Q23=33 Q22= 0
        COLUMNS  8- 9, 10-11 RESPECTIVELY

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q22 OUT OF RANGE: = 0
        COLUMNS  8- 9

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q23 OUT OF RANGE: =33
        COLUMNS 10-11

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q24A OUT OF RANGE: = 0
        COLUMNS 12-13

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q24A AND Q25A NOT CONSISTENT,
        COLUMNS 12-13, 14-15 RESPECTIVELY

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q26 OUT OF RANGE: = 0
        COLUMNS 16-17

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q27 OUT OF RANGE: =0
        COLUMN 18

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q28 OUT OF RANGE: =0
        COLUMN 19

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        Q29 OUT OF RANGE: = 0
        COLUMNS 20-21

ERROR: RECORD NO.=        4, CASE NO.=         1, ID NO.=    1, CARD NO.= 4
        0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
        .1 OUT OF RANGE: =0
```

```
       COLUMN 24

ERROR: RECORD NO.=       4, CASE NO.=       1, ID NO.=    1, CARD NO.= 4
       0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
       84 OUT OF RANGE: =0
       COLUMN 29

ERROR: RECORD NO.=       4, CASE NO.=       1, ID NO.=    1, CARD NO.= 4
       0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
       85 OUT OF RANGE: =0
       COLUMN 30

ERROR: RECORD NO.=       4, CASE NO.=       1, ID NO.=    1, CARD NO.= 4
       0000104003 3000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
       86 OUT OF RANGE: =0
       COLUMN 31

ERROR: RECORD NO.=       5, CASE NO.=       2, ID NO.=11083, CARD NO.= 1
       1108301020 0240880811 0222230000 0000000000 0000000000 0000000000 0000000000 0000000000
       ID NO. OUT OF RANGE: =11083
       COLUMNS 1-5

ERROR: RECORD NO.=       5, CASE NO.=       2, ID NO.=11083, CARD NO.= 1
       1108301020 0240880811 0222230000 0000000000 0000000000 0000000000 0000000000 0000000000
       AREA OUT OF RANGE: = 0
       COLUMN 10

ERROR: RECORD NO.=       6, CASE NO.=       2, ID NO.= 1083, CARD NO.= 2
       0108302024 0319235033 0200011114 3133455354 4444536111 1000001211 3114000000 1000000000
       ID NOT SAME AS ON PREVIOUS CARD.
       COLUMNS 1-5
       IMAGE OF PREVIOUS CARD 1:
       1108301020 0240880811 0222230000 0000000000 0000000000 0000000000 0000000000 0000000000

ERROR: RECORD NO.=       7, CASE NO.=       2, ID NO.= 1083, CARD NO.= 3
       0108303677 7677677677 7777676000 0001000200 0048101111 1000000000 0000000000 0000000000
       ID NOT SAME AS ON PREVIOUS CARD.
       COLUMNS 1-5
       IMAGE OF PREVIOUS CARD 1:
       1108301020 0240880811 0222230000 0000000000 0000000000 0000000000 0000000000 0000000000

ERROR: RECORD NO.=       7, CASE NO.=       2, ID NO.= 1083, CARD NO.= 3

       0108303677 7677677677 7777676000 0001000200 0048101111 1000000000 0000000000 0000000000
       Q19AA  OUT OF RANGE FOR COMMUNITY WITH NOISE PROGRAM: =0
       COLUMN 3[illegible]

ERROR: RECORD NO.=       7, CASE NO.=       2, ID NO.= 1083, CARD NO.= 3
       0108303677 7677677677 7777676000 0001000200 0048101111 1000000000 0000000000 0000000000
       Q19A(A-D)  NOT 8 FOR COMMUNITY WITH NOISE PROGRAM   Q19A(A-D)=0  0  0  0
       COLUMNS 39-42

ERROR: RECORD NO.=       8, CASE NO.=       2, ID NO.= 1083, CARD NO.= 4
       0108304231 3050105140 3302103022 1000000000 0000000000 0000000000 0000000000 0000000000
       ID NOT SAME AS ON PREVIOUS CARD.
       COLUMNS 1-5
       IMAGE OF PREVIOUS CARD 1:
       1108301020 0240880811 0222230000 0000000000 0000000000 0000000000 0000000000 0000000000
```

```
TTITUDINAL SURVEY DATA CLEANING MESSAGES                                                  DATE 092679       PAGE    13

ERROR: RECORD NO.=        9, CASE NO.=        3, ID NO.= 1063, CARD NO.= 1
       0106301020 0240881011 0222720000 0000000000 0000000000 0000060000 0000000000 0000000000
       AREA OUT OF RANGE: = 0
       COLUMN 10
       TOTAL NO. CF CARDS IS NOT MULTIPLE OF  4

****** END-OF-FILE ENCOUNTERED ******
       TOTAL CARDS READ =     9

******     PROGRAM    ENDS      ******
@ECF
@EOF IGNCRED - IN CONTROL MODE
@BRKPT PRINT$

END ONSITE PRINTOUT ON SEPTEMBER 26, 1979 AT 10:08:35
CLEANTEST*STEVEOUT(1).
```

```
*** ATTITUDINAL SURVEY DATA CLEANING BEGINS ***
PROGRAM WRITTEN BY:  DR. RAGU SINGH, CSC
LAST EDIT BY:  STEVEN G. EAGLESON, CSC, FALLS CHURCH, VA.
         ********VERSION 2.4(9)**********

      FLEXIBLE DATA RANGES:
      LOID=     1  HIID= 1500
      LONZ=     1  HINZ=   17
      LOAR=     1  HIAR=    7
      LOCLLS=   1  HICLLS=  100
      CITIES:   88      88      86      88
      MAX INT DATE=   60
      MAX NO OF ERRORS=     9

      WITHOUT COMMUNITY NOISE PROGRAM
      AFFECTS Q18, Q19 Q19A, Q21

ERROR: RECORD NO.=        5, CASE NO.=         2, ID NO.= 1111, CARD NO.= 1
       0111101011 0110681111 1111230000 0000000000 0000000000 0000000000 0000000000 0000000000
       ID SAME AS PREVIOUS CASE.
       COLUMNS 1-5
ERROR: RECORD NO.=       12, CASE NO.=         3, ID NO.= 1113, CARD NO.= 4
       0111304231 3999999999 9951999999 9000000000 0000000000 0000000000 0000000000 0000000000
       SEX IN COLS 24 OF CARDS 1 AND 4 NOT SAME.
       IMAGE OF PREVIOUS CARD 1:
       0111301011 0110381111 1112230000 0000000000 0000000000 0000000000 0000000000 0000000000
ERROR: RECORD NO.=       16, CASE NO.=         4, ID NO.= 1114, CARD NO.= 4
       0111404231 3999999990 1951999999 9000000000 0000000000 0000000000 0000000000 0000000000
       TOT.NO. ADULT RESIDENTS, COL 22,  OF CARD 1 LARGER THAN 029, COLS 20-21, OF CARD 4
       IMAGE OF PREVIOUS CARD 1:
       0111401011 0110681111 1511230000 0000000000 0000000000 0000000000 0000000000 0000000000
ERROR: RECORD NO.=       20, CASE NO.=         5, ID NO.= 1115, CARD NO.= 4
       0111504231 3999999999 9951999999 9000000000 0000000000 0000000000 0000000000 0000000000
       AGES ON CARDS 1 AND 4 NOT SAME.
       COLUMNS 24-26, 8-9 RESPECTIVELY
       IMAGE OF PREVIOUS CARD 1:
       0111501011 0110881111 1111650000 0000000000 0000000000 0000000000 0000000000 0000000000
ERROR: RECORD NO.=       25, CASE NO.=         7, ID NO.= 1117, CARD NO.= 1
       0111700011 0110881111 1111230000 0000000000 0000000000 0000000000 0000000000 0000000000
       CARD NO. WRONG OR OUT OF SEQUENCE.
       CARD FIELD VALUE = 0 COLUMNS 6-7
ERROR: RECORD NO.=       25, CASE NO.=         7, ID NO.= 1117, CARD NO.= 1
       0111700011 0110881111 1111230000 0000000000 0000000000 0000000000 0000000000 0000000000
       CARD NO.= 0 WHICH IS LESS THAN ONE.
******     PROGRAM     ENDS     ******
```

Reproduced from best available copy.

APPENDIX D RUNSTREAM EXPLANATIONS

Note on punch card codes:

The Univac 1110 computer uses FIELDATA as its character representation code. The FIELDATA punch card representation of a character is sometimes different from the standard IBM 029 representation (EBCDIC). If the user's communication software will not perform the translation from EBCDIC to FIELDATA, it will be necessary to punch the cards according to the FIELDATA format.

The user should refer to Table D-1, pick out the character and its FIELDATA card code and then find which character the FIELDATA card code corresponds to in the EBCDIC code. This last character is what the user will punch on the card. The three most important cases are given below:

FIELDATA Character	FIELDATA Code	EBCDIC Character Punched to Generate this Code
@	7-8	"
(	0-4-8	%
)	12-4-8	<

TABLE D-1 COMPARISON OF FIELDATA AND EBCDIC

Fieldata Octal	Fieldata Code	Fieldata Punched Card Code	EBCDIC (9000) Punched Card Code
00	@	7-8	4-8
01	[	12-5-8	12-8-2
02	]	11-5-8	11-8-2
03	#	12-7-8	3-8
04	Δ	11-7-8	11-7-8
05	SP	NO PUNCH	NO PUNCH
06	A	12-1	12-1
07	B	12-2	12-2
10	C	12-3	12-3
11	D	12-4	12-4
12	E	12-5	12-5
13	F	12-6	12-6
14	G	12-7	12-7
15	H	12-8	12-8
16	I	12-9	12-9
17	J	11-1	11-1
20	K	11-2	11-2
21	L	11-3	11-3
22	M	11-4	11-4
23	N	11-5	11-5
24	O	11-6	11-6
25	P	11-7	11-7
26	Q	11-8	11-8
27	R	11-9	11-9
30	S	0-2	0-2
31	T	0-3	0-3
32	U	0-4	0-4
33	V	0-5	0-5
34	W	0-5	0-6
35	X	0-7	0-7
36	Y	0-8	0-8
37	Z	0-9	0-9
40	)	12-4-8	11-8-5
41	-	11	11
42	+	12	12-8-6
43	<	12-6-8	12-8-4
44	=	3-8	6-8
45	>	6-8	0-6-8
46	&	2-8	12
47	$	11-3-8	11-3-8
50	*	11-4-8	11-4-8
51	(	0-4-8	12-5-8
52	%	0-5-8	0-4-8
53	:	5-8	2-8
54	?	12-0	0-7-8
55	!	11-0	12-8-7
56	,	0-3-8	0-3-8
57		0-6-8	0-8-2
60	0	0	0
61	1	1	1
62	2	2	2
63	3	3	3
64	4	4	4
65	5	5	5
66	6	6	6
67	7	7	7
70	8	8	8
71	9	9	9
72	'	4-8	5-8
73	;	11-6-8	11-6-8
74	/	0-1	0-1
75	.	12-3-8	12-3-8
76	⌑	0-7-8	8,7
77	–	0-2-8	0-5-8

Explanation of Figure 3-1, First System Initiation Runstream

@RUN statement, required, sets job run priority to routine (R), core limit at 25 kWords, job run time limit to 30 seconds, and maximum number of print pages to 20.

@PASSWD statement must be included only for batch jobs entered by card reader, otherwise it is optional.

@DELETE deletes old disk files from system. The C option indicates that this job is not terminate if there is no old file to delete. The delete will not affect files saved on tape. These deletes are needed because if the file already exists, and one tries to @CAT it again, the job will terminate with an error.

@CAT statements enter (catalog) the file names into the system and reserve space on disk storage. Note that the system will assume the full form of each file name to be proj-id*file-name. unless the user specifies a different qualifier.

@FIN ends the job, and causes the print out and accounting data to be sent to the remote-site at which the job was entered.

Explanation of Figure 3-2, Second System Initiation Runstream

The required @RUN card specifies the job run priority as routine. Core limit is set at 25 kWords, the run time limit is set at 4 minutes, and a hundred and fifty pages may be printed before the job will halt early.

@PASSWD statement is required only for batch jobs entered on with a card reader. For jobs submitted through a terminal this card is optional.

@DELETE,C will eliminate any old RAWDATA. file, if it should exist from a previous job, without terminating this run if it doesn't exist. This statement will insure that the next command, the @FILE, will not abort the job, as it would do if RAWDATA. did exist.

The @FILE,CR is used to create and initialize a new data file on disk. The C option indicates that the file will be cataloged only when the job terminates normally. The R option specifies that this is to be a 'read-only' file, which cannot be written to. (It is possible to clear this protection feature when necessary).

All the data cards from the survey should follow. These will be put onto the disk file.

The @ENDF signals the end of the data to the processor called by the @FILE.

The @FIN indicates the end of the runstream. Any print out is sent to the remote site along with accounting information.

Explanation of Figure 3-4, Sample CLEAN Runstream

The required @RUN statement specifies that the job run priority is routine (R), with 13 kWords of core. It limits the job to 4 minutes of computer time and 500 pages of output.

The @PASSWD statement is required for batch jobs entered through a card reader. Otherwise, it is optional.

The @ASG,A assigns an existing file for the exclusive use of this run. In this case the ASDAS system program file, LISTEN*ATTITUDINAL., and the user created proj-id*RAWDATA. files are allocated for the run.

The @USE statement links the proj-id*RAWDATA. file to the FORTRAN unit number 10, the unit from which program CLEAN reads data.

The @XQT statement calls up the absolute binary element of CLEAN for execution and starts the program.

The CLEAN initialization card follows, with the format specified in Table 3-2. The runstream is accessed as the system default for the FORTRAN unit number 5.

@FIN ends the runstream, causing program output to be routed, along with accounting information, to the remote site.

118<

Explanation of Figure 3-5, Runstream to Copy RAWDATA. to GOODDATA. file

The required @RUN statement sets the job run priority to routine (R). The core requested is 25 kWords of memory. Thirty seconds of run time and 50 pages of output may occur before the job is terminated.

The @PASSWD statement is required when the job is submitted through a card reader. Otherwise, it is optional.

@CHG,Z will clear the read-only protection from GOODDATA. so that writing can be done on it.

@ERS GOODDATA erases the file's old data.

With the @ASG,A two existing files, GOODDATA. and RAWDATA., are exclusively assigned for this run's use.

@COPY RAWDATA.,GOODDATA. duplicates the contents of the RAWDATA. file on GOODDATA.

@FREE releases the exclusive control of GOODDATA. This is required so that the next command can successfully operate.

@CHG,V sets the file protection to read-only mode.

@FIN terminates the job and routes print out and accounting to the job's entry site.

Explanation of Figure 3-6, Sample Edit Runstream

The @RUN sets the job priority as routine (R), and requests 35 kWords of core, 3 minutes of run time and 300 pages of output.

The @PASSWD statement is required only for batch jobs entered through a card reader. Otherwise, it is optional.

@CHG,Z releases the read-only protection from RAWDATA. so that changes can take place.

@ASG assigns RAWDATA. for the exclusive use of this run.

@US*ER.ED calls up the University of Maryland file editor, an enhanced version of the Univac @ED processor.

A blank record puts the editor into 'edit mode' from the initial 'input mode' condition which the editor enters automatically when it finds an empty file.

The editor directives, as described in Section 3.2.3.1, are to follow. The last command is EXIT, which will cause the editor to replace the old RAWDATA. file with the new, edited version.

@FREE releases RAWDATA. so that the following @CHG can operate on it.

@CHG,V sets the file protection to read-only mode.

@FIN ends the job, routing print-out and accounting information to the site of submission.

Explanation of Figure 3-9, Sample VARIANCE Runstream

The @RUN statement sets the job run priority at routine (R), with 64 kWords of core, and a limit of 5 minutes of computer time, and 50 pages of printed output.

@PASSWD is required for batch jobs entered with a card reader, otherwise it is optional.

@CHG,Z removes the read-only protection from VARIANOUTPUT.

@ERS erases the old contents of the VARIANOUTPUT. file.

@ASG,A is used to assign VARIANOUTPUT., LISTEN*ATTITUDINAL. and GOODDATA. files for the exclusive use of this run.

The @BRKPT statement shunts all of the job's printed output away from the temporary PRINT$ file (that exists only as long as the run) to VARIANOUTPUT., thus saving the program's output for later reuse. Note that the non-standard syntax of this command requires that no period follow the file name (VARIANOUTPUT).

@XQT calls up the FORTRAN program VARIANCE which will process the GOODDATA. file and generate a listing of results.

This @BRKPT PRINT$ statement reverses the effects of the previous @BRKPT, so that the accounting information will be routed to the submittal site in a normal manner.

@FREE will release the VARIANOUTPUT. file so that the @CHG following can operate on it.

@CHG,V sets up read-only protection for the VARIANOUTPUT. file.

The @SYM statement will route a listing of VARIANOUTPUT. to the specified site-id. Multiple copies (1 to 63) are specified by inserting a number between the two commas on this card. The U option specifies that VARIANOUTPUT. is not to be deleted once the @SYM process is completed (which will occur if the U is omitted).

@FIN ends the run and routes accounting information to the submittal site.

Explanation of Figures 3-11, 3-12, 3-13, 3-13a, 3-14, 3-15, 3-16, Runstreams to run TAILOR

The @RUN statement specifies routine (R) priority, requests 35 kWords of memory, and limits run time to 3 minutes with no more than 30 pages of output.

The @PASSWD is required only for batch jobs submitted with a card reader, otherwise it is optional.

@CHG,Z clears the read-only protection from the user's proj-id*ATTITUDINAL. file.

The @ASG assigns 4 files for the exclusive use of this run. These are: LISTEN*ATTITUDINAL., proj-id*ATTITUDINAL., proj-id*TAILORINPUT., and proj-id*TAILOROUTPUT.

The @USE statements link up the FORTRAN unit numbers in TAILOR to the files assigned to this run. TAILOR reads from unit 4 and writes to unit 7.

Since the @COPY command cannot access the individual program elements of the LISTEN*ATTITUDINAL. file, the @US*ER.ED file editor is used to make the transfer instead. TAILORINPUT. is specified as the file being edited, the ADD command copies a source program from LISTEN*ATTITUDINAL. The EXIT command releases the editor's control.

The @XQT statement cause TAILOR to begin execution. Like CLEAN, TAILOR will read from FORTRAN unit 5, which is the NCC default for the runstream. At least 1 record with the TAILOR control flags must follow, as described in Section 3.4. Any SPSS source language statements to be inserted must follow the flag record, as shown in Figure 3-13a.

@EOF signals 'end of file' to TAILOR.

The @COPY,I can be used to take the TAILOROUTPUT. data file and include it as a program element in the proj-id*ATTITUDINAL. file. As noted previously, the reverse case is not possible, necessitating the use of the *US*ER.ED editor.

The @FREE releases the user's ATTITUDINAL. file so that the next command can operate properly on it.

@CHG,V restores the read-only protection to the ATTITUDINAL. file.

@FIN ends the runstream and routes the job print out and billing information to the site of submission.

Explanation of Figure 3-19, Sample RECODE1 Runstream

The required @RUN statement sets the job run priority to routine (R), requests 64 kWords of memory, and limits run time to 5 minutes, with a maximum of fifty pages printed output.

An @PASSWD statement is needed for batch jobs submitted through a card reader. Otherwise, it is optional.

@CHG,Z is used to remove the read-only file protection from two files, SYSFILE1. and ZSCORES.

@ERS erases the old contents of the above files.

The NCC SPSS utility program is assigned to this run. This is because this processor may have been 'rolled out' if it has not been recently used. The assignment insures that this job will not abort with a FILE ERROR type message. Files ATTITUDINAL., SYSFILE1., ZSCORES., and GOODDATA. are assigned to the run.

The @USE statements link the file name with the unit numbers required by the SPSS utility. Unit 4 is written to since a SAVE FILE or ARCHIVE FILE is used in the program. Unit 8 is the data input and Unit 20 is the Z-Scores output.

The statement @LIB*SPSS7-2.SPSS starts the SPSS package execution.

The @ADD command has the effect of adding to the runstream, at the location of the @ADD statement, all of the data and/or control statements in the specified file element. Thus, this is the equivalent of having the SPSS program in ATTITUDINAL.RECODE1 on punched cards following the @LIB statement.

@EOF insures that the SPSS processer gets an end of file condition once the SPSS program is completely processed.

SYSFILE1. and ZSCORES. are @FREEed so that the @CHG,V can successfully restore the read-only file protection.

@FIN ends the job, routing print-out and billing information to the site of job submission.

Explanation of Figure 3-20, Sample RECODE2 Runstream

The @RUN statement sets the job run priority to routine (R), reserves 64 kWords of memory and limits run time to 5 minutes with a maximum of 40 pages of print out.

The @PASSWD statement is required for batch jobs submitted through a card reader, otherwise it is optional.

@CHG,Z removes the read-only protection from file SYSFILE2.

@ERS erases the old contents of SYSFILE2.

@ASG,A assigns existing files to this job. These are: proj-id*ATTITUDINAL. (filled by TAILOR), LIB*SPSS7-2. (the SPSS package file), SYSFILE1., SYSFILE2. and ZSCORES. Note that the SPSS file is assigned so that it is sure to be available when it is called up for execution.

The @USE statements link the assigned files to the unit numbers that SPSS requires. Unit 3 is for the SPSS 'GET FILE' statement. Unit 4 is for the SPSS 'SAVE FILE' or 'SAVE ARCHIVE' statement. Unit 8 is reserved as input for raw-data-input files.

The @LIB*SPSS7-2.SPSS calls up the statistical package for execution.

The @ADD statement has the effect of adding to the runstream the contents of the specified file element, RECODE2. Thus, this is the equivalent of having the SPSS program RECODE2 on punch cards following the above @LIB.

@EOF insures that the SPSS package gets an end of file condition once it has finished with RECODE2.

@FREE releases the SYSFILE2. file from the exclusive control of this job, so that the following command will function properly.

@CHG,V restores SYSFILE2. to read-only status.

@FIN ends the job, causing print-out and accounting information to be routed to the site of submission.

Explanation of Figure 3-21, Sample FREQUENCIES Runstream

The @RUN statement sets the job run priority to routine (R), requests 64K words of memory and limits run time to 8 minutes, with a maximum of 600 pages of output.

@PASSWD is required for batch jobs entered on a card reader, otherwise it is optional.

@CHG,Z removes the read-only protection from file FREQOUT.

@ERS erases any old contents of FREQOUT.

The @ASG assignment of LIB*SPSS7-2., the system SPSS utility, is done to insure that it is available for use, not rolled out.

Files SYSFILE 2., ATTITUDINAL. and FREQOUT. are assigned to the run.

@USE links up the unit number that SPSS requires for the 'GET FILE', unit 3, to SYSFILE2.

@BRKPT diverts the job's printed output to the FREQOUT. file. PRINT$ is a temporary file created for each job that disappears once the @FIN routes it for printout. Note the syntax of this statement required that the normal period after FREQOUT should be omitted.

@LIB*SPSS7-2.SPSS calls up the system's SPSS utility for execution.

The @ADD command has the effect of adding to the runstream the contents of the specified file element, ATTITUDINAL.FREQUENCIES. Thus, this is the equivalent of having the SPSS program in the file element on punch cards following the above @LIB statement.

@EOF insures that the SPSS package gets an end of file condition once the contents of ATTITUDINAL.FREQUENCIES has been used.

@BRKPT PRINT clears out the previous @BRKPT settings, so that the accounting information is routed in the normal manner.

@FREE releases FREQOUT. from the exclusive control of this run, so that the @CHG can operate on it.

@CHG,V restores the read-only file protection to FREQOUT.

@SYM,U sends a copy of the FREQOUT. file to the specified site-id. The ,U option is required to keep FREQOUT. from being deleted after this process is complete.

@FIN ends the job and routes job accounting to the site where the job was entered.

Explanation of Figure 3-22, Sample Runstream for CROSSTABS

The @RUN statement sets the job run priority to routine (R), reserves 64K words of memory, limits run time to 10 minutes, and gives the maximum number of pages as 600.

@PASSWD is required for batch jobs entered through a card reader. Otherwise it is optional.

@CHG,Z removes the read only file protection from XTABSOUT.

@ERS erases any old contents of the file XTABSOUT.

The system utility LIB*SPSS7-2. is assigned with the @ASG,A so that it is sure to be available when it is called up for execution. The other files that are assigned are SYSFILE2., ATTITUDINAL., and XTABSOUT.

The @USE statement links the unit number 3, which SPSS expects for input, to the file SYSFILE2.

@BRKPT diverts the job's print out to the XTABSOUT. file, away from the temporary PRINT$ file.

@LIB*SPSS7-2.SPSS calls up the system's SPSS processor for execution.

@ADD will add to the runstream the contents of the specified file element ATTITUDINAL.CROSSTABS. This is the equivalent to having the SPSS language statements in this element on punch cards following the above @LIB.

@EOF insures that the SPSS utility receives an end of file condition after it has finished with the CROSSTABS element.

@BRKPT PRINT$ restores the normal flow of output to the temporary PRINT$ file, so that the job accounting is handled in the normal manner. 128<

@FREE releases XTABSOUT. from exclusive control of the job, so that the following @CHG command can work.

@CHG,V restores read only protection to the XTABSOUT. file.

@SYM,U sends a copy of XTABSOUT. to the specified site-id. The ,U option indicates the normal delete action of the @SYM is not to take place.

@FIN end the job, and routes accounting information to the job submission site.

Explanation of Figure 3-23, Sample REGRESSIONS Runstream

The @RUN statement sets the job run priority to deferred (D), requests 64K words of memory. Runtime is limited to 20 minutes, with a maximum of 600 pages printed output.

@PASSWD is required for batch jobs entered on a card reader, and is optional otherwise.

@CHG,Z removes the read-only protection from the REGRESSOUT. file.

@ERS erases any old data in REGRESSOUT.

@ASG,A assigns the system's SPSS utility (LIB*SPSS7-2.) to this run, insuring that it is available for use. Files SYSFILE2., ATTITUDINAL., and REGRESSOUT. are also assigned.

@USE links the unit number 3, the unit from which SPSS expects input, to the file SYSFILE2.

@BRKPT diverts the run's printout away from the temporary PRINT$ file to the cataloged (permanent) REGRESSOUT. file. Note that the syntax requires the absence of the normal period usually found after the file name.

@LIB*SPSS7-2.SPSS calls up the system's SPSS processor for execution.

@ADD will add to the runstream the contents of the file element ATTITUDINAL.REGRESSIONS, acting as if its contents follow the @LIB statement above.

@EOF insures that the SPSS utility receives the end of file condition once the REGRSSIONS element is finished.

@BRKPT restores the normal flow of print out to the temporary PRINT$ file, so that the billing information is handled normally.

@FREE releases REGRESSOUT. from the exclusive control of this run, so that the next command will operate on it.

@CHG,Z restores the read-only protection to the REGRESSOUT. file.

@SYM,U sends a copy of the REGRESSOUT. file to the specified site-id. The ,U option inhibits the @SYM's normal action of deleting the file once the copy has been printed.

@FIN ends the job, routing print-out and biling information to the site of submission.

Explanation of Figure 3-24, Runstream to Archive ASDAS Results

The required @RUN statement will set the job run priority to routine (R), with a requested 35K words of memory, one tape drive unit, and a run time limit of 5 minutes.

The @PASSWD statement is required for batch jobs entered with a card reader and is otherwise optional.

@DELETE,C removes the specified tape name from the system. The C option indicates that the job is to continue (not error abort) if there is no existing file name to delete. Removing the file here insures that the next command will always work. This procedure does not affect the contents of the tape.

@ASG,U assigns the tape reel indicated by tape-num to the newly created file name/tape name as specified by the ,U option. The P designates the file as public, the /W is to allow writing to the tape, //ASDAS specifies a write-key of 'ASDAS', and 36N indicates a nine track tape.

@USE sets up a logical link between the tape file and the name OUTTAPE. so that references to OUTTAPE. actually result in access to the named tape file. This is done to make the runstream more self explanatory.

The ASDAS program and data files are assigned to this runstream.

@COPOUT (with no options specified) will copy the entire contents of the ATTITUDINAL. program files to the tape file.

@COPY is used to duplicate on the tape the data files created by running the ASDAS system.

132<

@REWIND performs a rewind on the tape file so that the next command can find the files just copied to it.

@US*ER.FLIST is a utility processor that produces file listings, the ,B option is for the brief format giving the name and type of each file on OUTTAPE.

@FIN ends the job, routing print out and accounting data to the site where the run was submitted.

Explanation of Figure 3-25, Runstream to De-Archive ASDAS Results

The @RUN sets up a routine (R) priority job that requires 35K words of memory and one tape drive. Five minutes of CPU time can be used before the run will terminate.

@PASSWD is required for batch jobs submitted with a card reader. Otherwise it is optional.

@DELETE is used to delete the specified files if they currently exist. The C option allows the run to continue if the files do not exist. Deleting the files allows the following statements to operate successfully every time, preventing job error termination.

@CAT catalogs the specified files as disk storage files. The @ASG,U assignment is not done in this case since the following COPY processors do an automatic assign anyway.

The @DELETE,C removes the tape file name from the system, if it exists, so that the ,U option on the next command will not halt the job abnormally.

@ASG,U creates a new tape file name, and links it to the tape reel number indicated. 36N specifies a 9 track tape. No write-key is provided in this assign, since the job only requires that the tape be read.

@USE logically links the tape file name to the designation INTAPE. Any reference to INTAPE. from this point on will access the tape file, allowing a more descriptive runstream.

@REWIND insures that the tape reel is at the starting point.

@COPIN copies the program files from the tape to the indicated disk files.

@COPY copies the data files from the tape to the disk files specified. Note that the order of the files on the tape as it was written (see Figure 3-24) must be preserved if the disk file names are to match their contents.

@FREE releases the tape drive and tape reel from exclusive use of this run since the copying is done. Omitting this will unnecessarily tie up the tape drive until the run is over.

@HDG sets up a print page header as specified. Since no options are specified date and page numbers are automatically added as well.

@PRT,T lists in readable form the 'table of contents' that the system uses to keep track of program files. The program element names and pertinent information are printed. Note that for batch jobs the long or full format will be printed. In demand processing jobs the brief format will be used unless the L option is also added.

A new @HDG heading is set up, and @PRT,T is used to list the 'table of contents' for the second program file.

@HDG,N will reverse the results of the previous @HDG, so that printout is no longer given page headings automatically.

@SYM creates listings of the indicated files printed separately from this runstream's printout at the indicated site-id. The U option inhibits the normal delete action of @SYMing.

@FIN ends the job and routes printout and accounting information to the site of job submission.

Explanation of Figure 3-26, Sample Runstream to Delete LISTEN System

@RUN sets up a routine (R) priority job and requests 35K words of memory. CPU processing time is limited to 30 seconds and 20 pages of printout is specified as the maximum.

@PASSWD is required only for batch jobs entered with a card reader, and is optional in all other cases.

@DELETE removes all of the user created disk files from the system, releasing the space for other use.

@FIN ends the job and routes printout and accounting information to the site where the job was submitted.

APPENDIX E UNIVAC ECL SUMMARY

This section is intended to provide a brief description of those Executive Control Language (ECL) commands which are most frequently used. It is not intended as an ECL Manual. For additional information the user should consult the EPA National Computing Center User Reference Manual (8), Volume VII. Table E-1 lists the commands which are discussed in this section. Figure E-2 describes the conventions used in presenting these commands.

E.1 RUN CONTROL STATEMENTS

This subsection contains discussions of the most commonly used statements used in job control.

E.1.1 The @RUN Statement

The @RUN identifies a run and provides to the system certain parameters necessary for scheduling and accounting purposes. The format of the @RUN Statement is:

@RUN, <p> </o>b̸ <runid></core></tapes></disk>,[account] /<userid>, <project></class>

Where<p>=priority defined as follows:

P = preferred	highest priority/cost
R = routine	standard priority/cost
D = deferred	lowest priority/cost
b̸ = routine	default

<o> = options
  S = run is to be sequenced after previous run from the input device
  R = restart run in the event of a failure
<runid> = 1-6 alphanumeric characters
<core> = maximum core required in decimal thousands of words (defaults: batch - 45k, demand - 65k)
tapes = maximum number of tape drives assigned concurrently (defaults: batch - 3, demand - 2)
<disks>= not currently used
[account] = must be a valid account number
<project>= used by the system as a default qualifier for filenames
<class> = class where
  P = production
  D = development
  N,b̸ = unspecified

137<

runtime = maximum suptime required by run in minutes (default = 2)

pages = output page estimate (default = 100, maximum = 99999)

start-time = specifies delay time after which run will be considered for execution in hhmm; an input of Dhhmm specifies the wall clock time at which run will be considered for execution

Table E-1. ECL Summary

Type	Command	Page	Description
Run Control	@RUN	E-1	Initiates a run control stream
	@FIN	E-4	Terminates a run control stream
	@ADD	E-4	Inserts data or control statements from another file
	@START	E-5	Schedules a batch job from within an executing job (demand or batch)
File Control	@ASG	E-5	Assigns a mass storage file for subsequent use by the run
	@FREE	E-7	Frees an assigned file
	@USE	E-8	Equates an external filename to an internal, or simply a shorter filename
	@SYM	E-8	Outputs a print file to an on-site or remote printer
	@QUAL	E-9	Defines a default filename qualifier
	@BRKPT	E-9	Switches standard output unit to a user specified file
	@XQT	E-10	Executes an absolute element
	@DATA	E-11	Allows user to build a data file
	@ELT	E-12	Allows user to build a program file element
	@END	E-13	End-of-data card for @Data or @ELT statement
Utility	@COPY	E-14	Copies one file/element to another
	@DELETE	E-15	Deletes a file from the catalogue
	@CHG	E-16	Changes the characteristics of a file

Table E-1. ECL Summary (Continued)

Type	Command	Page	Description
Utility	@PRT	E-16	Prints a file (or an element) or its directory
	@MOVE	E-16	Moves a tape file past an end-of-file mark
	@REWIND	E-16	Rewinds a tape file

E.1.2 The @FIN Statement

The @FIN Statement signals that the end of a run has been reached. It must be the last statement of each run control stream. The format of this statement is:

```
@FIN
```

When a @FIN statement is encountered, the Executive accounting routines are initiated in order to produce the run session statistics.

E.1.3 The @ADD Statement

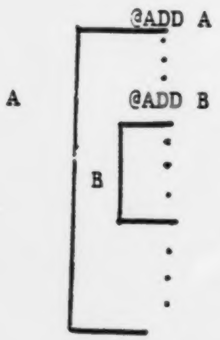
The @ADD Statement provides the capability of inserting ECL statements and data images from a file into the run control stream. The file must be in the system data format (SDF); i.e., must have been created by the @US*ER.ED, @DATA, @FILE or @ELT statements or a user program. The format of this command is:

```
@ADD,optionsb̸filename
```

where options =

- D - allows insertion of files or elements under the DATA or ELT processors.
- E - return control as if an @EOF had been encountered when the end of the file is reached.
- L - lists all control statements encountered in the added file (demand only).
- P - the @ADD statement is to be printed in the output listing.

Following the end of the added file the control stream is automatically resumed at the image following the @ADD statement. @ADD files may be nested to a level of 63, i.e.,



The user must be careful not to use statements in an inner @ADD which alter the content of an outer @ADD.

E.1.4 The @START Statement

The @START statement allows the user to submit a run from within an operating runstream (batch or demand). The simplest format of this statement is:

```
        @START,<p>/<o>b̸<filename><,set>
```

where<p>= priority as discussed in E.1.1
<o>= options as discussed in E.1.1
<filename>= name of a SDF file or element created by the @DATA, @FILE, @ELT or @US*ER.ED processors.
<set>= contains an octal number to which the condition word of the run to be started will be initially set

If a user wishes to change any parameters of the @RUN Statement contained in the file, a more complex version of the statement may be used:

```
@START,<p>/<o>b̸<filename><,set></siteid><,runid><,account><,project><,runtime>
     <,pages></cards><,start time>
```

where each parameter is defined above or in paragraph 9.1.1.

E.2 FILE CONTROL STATEMENTS

This section discusses the most commonly used file control statements.

E.2.1 The @ASG Statement

The @ASG statement is used to name an external file to be assigned to the requesting run. The statement has two basic formats, depending on device type:

- Mass storage -

```
        @ASG,<o>b̸<filename><,type></reserve></granule></maximum>
```

where o = options as follows:

C - specifies that the file is to be catalogued only if the run terminates with no errors.

U - specifies that the file is to be catalogued regardless of manner of run termination.

R - specifies that the file is to be catalogued as a read-only file.

P - specifies that the file is to be catalogued as a public file, i.e., any run may access the file.

D - specifies that the file is to be decatalogued only if the run terminates with no errors.

K - specifies that the file is to be decatalogued regardless of the manner of termination.

X - specifies that the run is to have exclusive use of the file.

A - specifies that the file is currently catalogued.

Y - specifies that the file is assigned solely for the purpose of examining the master file directory.

T - specifies that a temporary file is to be created for this run only.

filename = has the format

qualifier*file(cycle)/r-key/w-key

The components of the filename are described in detail in the Executive System Programmer's Reference Manual. The following fields are ignored if the file is currently catalogued:

type = device type (See NCC Users Guide[8])
reserve = number of granules to be reserved for the file
granule = granule size
TRK (one track, 1792 words)
POS (one position 114, 688 words)
maximum = maximum number of granules to be assigned to the file

o Magnetic tape

@ASG,o/subb̸<filename>,type/units/log/noise,reel$_1$/reel$_2$/...

where o = options from the following:

C, U, R, P, D, K, A, T as defined above.

sub = W - specifies that a write ring is to be placed in the tape when mounted

Parity
E - even
O - odd

B - binary (no translation)

I - translate Fieldata to BCD during write, BCD to Fielddata during read

type= T - specifies tape, type independent
16 - 7 track tape
36N - 9 track tape

units = integer specifying the number of units

log = single character to indicate a logical channel

noise = decimal constant specifying minimum frame count to be read

reel$_1$/reel$_2$/ ... = specifies reel numbers (assigned by the tape librarian) in the order to be used.

E.2.2 The @FREE Statement

The @FREE statement de-assigns a file when no longer needed by a run. The format of the command is:

```
@FREE,ecfilename
```

where the following options are available:

A - Release only the @USE-attached name specified in filename field. Take no other action unless indicated by other options.

B - Release the @USE-attached name specified in filename field. If this is the only @USE-attached name, FREE the file.

D - Drop this catalogued file from the master file directory. In order for this to be accomplished, the file must be assigned with valid keys, if keys exist in the master file directory.

I - Inhibit cataloguing of a file that was assigned using @ASG,C or @ASG,U.

R - Do a @FREE, but do not release any @USE-attached names (unless the A option is also used). Save all pertinent information to reestablish the assignment.

S - Do a @FREE of the file but retain the physical tape unit.

X - Remove this run's 'exclusive use' lock from an assigned file. Take no other action unless indicated by other options.

A file that is named on a @FREE statement can no longer be referenced by the run; it can, of course, be re-established by an @ASG statement, provided its facility requirements can be met.

E.2.3 The @USE Statement

The @USE control statement enables the user to refer to any particular file by two or more names. The need for the additional names arises from three conditions:

1. Simplify run construction by allowing the equating of an 'external-filename' to a shorter 'internal-filename'.

2. Resolve identical filename portions to 'external-filename' or 'attached' names.
3. Connect names coded into programs to 'external-filename' or 'attached' names.

The format of the @USE statement is as follows:

```
@USEb̸<internal-filename>,<external-filename>
```

where 'internal-filename' is the 12 character name by which the file is referred to within the run or after the @USE statement in the control stream and 'external-filename' is the name under which the file is to be assigned or is assigned (and possibly catalogued). The 'external-filename' has the standard filename format.

E.2.4 The @SYM Statement

The @SYM statement provides the user with the capability of selecting a symbiont, or class of symbionts, to print or punch selected files. When a @SYM statement is encountered the specified file is entered into the specified symbiont queue. Any file which is to be printed or punched must be scheduled for processing with the @SYM statement. The format of the @SYM statement is:

```
@SYM,<o>b̸<filename>,<number>,<device-name>
```

Where o = a combination of the following:

A - Specifies all files on the tape are to be printed (punched) in the order they appear on the tape: therefore, the filename given on the @SYM statement must be a tape file.

C - Directs the file to the card punch at the remote site specified in the device field. If omitted implies printing when @SYMing to a remote site.

D - If this option is specified and the filename is the generic name PRINT$, the current part of the PRINT$ file is deleted at the time the file is closed and no output is produced.

U - Inhibits decataloguing of the file when processing is completed; applicable only to user-defined files.

filename -specifies the file to be processed and directs tape mounting required. Filename must be catalogued.

number -specifies the number of copies of output. This field is omitted if only one copy is required.

device-name -specifies the name of the print or punch device to which output is directed, (remote siteid or userid).

E.2.5 The @QUAL Statement

The @QUAL statement allows the user to specify a filename qualifier for implied usage on succeeding control statements involving filenames. The format of this statement is:

```
@QUALb̸qualifier
```

where:

qualifier is an alphanumeric field which precedes filenames separated by an asterisk (*).

The following is an example of the use of the @QUAL statement: @QUAL MIKE

A subsequent use of the filename

*CONTROL.

would be interpreted as

MIKE*CONTROL.

E.2.6 The @BRKPT Statement

The @BKRPT statement is used to close output symbiont files. The symbiont files closed by the @BRKPT statement must be the normal print and punch files produced by PRINT$ and PUNCH$ or alternate print files produced by PRINTA$ and PUNCHA$. The @BRKPT statement has three basic formats as follows:

- o @BRKPTb̸PRINT$
- o @BRKPTb̸PUNCH$

This format of the @BRKPT statement causes the current normal print of punch file to be closed and queued for printing or punching and starts a new normal print or punch file.

- o @BRKPTb̸PRINT$/filename
  @BRKPTb̸PUNCH$/filename

This format of the @BRKPT statement causes the current normal print or punch type output to a file whose name is specified in the 'filename' field. The 'filename' must currently be assigned to the RUN with the Project-id as the file qualifier or no qualifier.

- o @BRKPTb̸filename

This format of the @BRKPT statement causes the alternate print or punch file specified in the 'filename' field to be closed.

If the alternate print or punch file was assigned by the system, the file will be freed and queued for printing or punching. If it was assigned by the user, a @FREE and @SYM statement must occur to print or punch the file, as shown in the following example:

```
@ASG,AƀOUTFIL.
@BRKPTƀPRINT$/OUTFIL
@US*ER.COPIE,CKƀINFILE.
@BRKPTƀPRINT$
@FREEƀOUTFIL
@SYM,UƀOUTFIL,,PR
```

E.2.7 The @XQT Statement

For executing an absolute program, the following control statement is used:

```
@XQT,<o>ƀ<filename>
```

o =Any 'options' specified are available to the user's program by the OPT$ Executive Request whenever it is initiated.

filename = 'filename.eltname/version'; specifying the absolute program to execute. The subfield 'filename' is the name of the file; 'eltname/version' identifies the element (and version, if any). In the absence of an absolute element, all of the relocatable elements in the file are both collected and executed (a relocatable element may be explicitly named on an @XQT statement, however).

E.2.8 The @DATA Statement

The @DATA Statement may be used to introduce standard format data files, found in the control stream, into the system for residence on a mass-storage device. A primary use for this feature is to allow the user to build data files which are actually whole or part of control streams. These files can then be called on by the @START statement to start an independent run, or by the @ADD statement for inclusion into the current run or a subsequent run. A data file correction feature is also available via the @DATA statement. The user can make a correction to an independent run stream and then @START it, or make corrections to a partial stream and then @ADD it to the run. The @DATA statement can, of course, simply be used as a convenient means of generating and maintaining a user data file, rather than a control-stream type file.

The format of the @DATA statement is:

@DATA,<o>b̸<filename$_1$><,filename$_2$>,sentinel

where o =

L - Produce a complete listing of the file.

U - Update. Produce a new file cycle specified by filenamel

I - Insert. Initial insertion of data into the data file specified in filenamel .

filenamel - specifies the name of the file to which the data images following the @DATA statement apply.

filename2 - specifies the updated file to be generated when no options are input.

sentinel - specifies a 1 to 6 character code used to match pairs of @END and @DATA statements. All images between matched pairs of sentinels will go into the data file being created.

If the 'I' option is omitted, and both <filenamel> and <filename2>are specified, the data images following the @DATA control statement are interpreted as corrections to<filenamel>. A new updated file identified by <filename2> is generated. If the 'options' and <filename2> fields are omitted, the 'L' option is assumed. <filenamel> is listed but no new file is generated. The 'U' option creates a new f cycle of <filename>.

E.2.9 The @ELT Statement

The @ELT control statement introduces an element into a particular program file from the control stream. It may also be used to make corrections to a symbolic element in a program file. The element or the corrections follow the @ELT statement in the control stream.

The format of the @ELT statement is:

@ELT,<o>b̸<filename$_1$><,filename$_2$>,sentinel

where o =

A	Absolute element
R	Relocatable element
S	Symbolic element
D	Data element
I	Insert. Initial insertion of an element into a program file
U	Update. Produce a new cycle of a symbolic element.
L	Produce a listing of the complete symbolic element.

The options 'A', 'R', 'S', and 'D' identify the element type. Types 'S' and 'D' are both considered symbolic elements and may be corrected in the same manner.

The @ELT statement indicates the element processor which operates in one of two modes. It inserts new elements into the program-file from the control stream and updates an element already in the program-file.

The <filename$_1$> field identifies the input element by filename, elementname, version and cycle (when appropriate).

When the 'I' option is specified, the element in the control stream is given the name specified in the <filename$_1$> field and inserted into the program-file specified in the <filename$_1$> field.

When the 'U' option is specified, the corrections in the control stream are applied to the element identified in the <filename$_2$> field, and a new cycle of the symbolic element is placed in <filename$_2$>, <filename$_1$> remains unchanged.

When the <filename$_2$> field is present and the 'U' option is not specified, the corrections in the control stream are applied to <filename$_1$> field and inserted into the program file specified in <filename$_2$>.

The 'L' option will produce a complete listing of a symbolic element. The 'L' option is not applicable for absolute or relocatable elements.

The 'data element' may contain control statements. Therefore, the data following the @ELT,D statement must be terminated with an @END statement with a sentinel exactly the same as found on the @ELT,D statement. The sentinel field need not be coded (blank sentinels). It is a six character field used to match up pairs of @END and @ELT statements. All images between matched pairs of sentinels will go into the data element being created. The @ELT,D statement may be used to insert @RUN or @ADD control streams into a program file as elements which may be called later by the @START or @ADD statement.

Element types 'A' and 'R' and 'S' are terminated by the next non-@EOF control statement in the control stream. They need no corresponding @END command.

E.2.10 The @END Statement

The @END control statement marks the end of the data that follows a @DATA or @ELT,D statement. The format of the @END statement is:

@ENDb̸<sentinel>

The sentinel field is optional. It is coded exactly the same as the corresponding field on a @DATA or @ELT,D statement when being used to bracket images of the data.

E.3 UTILITY ROUTINES

The NCC provides a set of standard File Utility Routines and Program Utility Routines (FURPUR) which consists of maintenance routines for files containing data or programs. This paragraph discusses some of the most frequently used commands.

E.3.1 The @COPY Command

The @COPY command copies an entire file by reading tape or mass-storage and writing to tape or mass-storage. @COPY manipulates both program and data files. The only partial file operation allowed is to add an element from one mass-storage program file to another mass-storage program file.

The format of the @COPY statement is:

@COPY,<o>b̸<filename$_1$><filename$_2$>

where:

<filename$_1$> =input filename or eltname
<filename$_2$> = output filename or eltname

and <o> =

None - <filename$_1$> is copied to <filename$_2$>

P - An entire program file is to be copied from filename$_1$ to <filename$_2$>(both of which are MASS-STORAGE files). Deleted elements are not copied (see @DELETE), and <filename$_2$> is packed. The table of contents for <filename$_2$> is updated. The entry point table of <filename$_2$> is not reconstructed and the file must be prepared by the @PREP statement if it is to be used in a LIB statement to the MAP processor.

The following 'options' are available for dealing with elements:

I - The input file named <filename$_1$> is source code data in System Data file Format (SDF). These data are to be added to a program file. The program file and the identification to be given to the new element are described in the <filename$_2$> as program file element/version. The symbolic element is given a cycle number of 0 when it is inserted into the program file.

S,R,A - These three options may be used singly or in combination for copying elements from one program to another: 'S' for symbolic, 'R' for relocatable, and 'A' for absolute.

E.3.2 The @DELETE Command

The @DELETE command may be used to delete one or more entries from the master file directory or one or more elements from a program file on mass-storage.

If an entire file is deleted from mass-storage, the storage area is released for reuse.

If a program file element is deleted, the element table entry is flagged but the physical storage area on mass-storage remains assigned until a @PACK command is found.

The @DELETE statement has the following format:

@DELETE, <o> ƀ <filename$_1$> , <filename$_2$> ,...<filename$_n$>

No options apply when deleting a catalogued file. The available options for program file element deletion are S,R, and A.

Several elements of the same type may be deleted by the same command. Each element must be described by name and must be from the same file. Version name should be included as needed to further identify the element.

Deletion of multiple elements in a file may be accomplished by additional specification fields to the command card (via @DELETE<filename$_1$,> filename$_2$...>) where <filename$_1$> is of the form filename.eltname/version and <filename$_2$> thru <filename$_n$> are of the form eltname2/version2., eltname3/version3,etc.

E.3.3 The @CHG Command

The @CHG command changes the name and security keys (read-key, write-key) of a catalogued file.

The format to change the name of a catalogued file is as follows:

@CHG,<o>ƀ filename

Note: @CHG for filenames is not currently implemented

See Volume 3, Section 4.2.15 of the UNIVAC Executive System Programmers Reference Manual for explanation of options.

E.3.4 The @PRT Command

The @PRT command prints information about a file on the contents of a source element of a program file.

This control statement does not list absolute or relocatable elements. The @PRT statement has the following general format:

@PRT,<o>ƀ<filename$_1$><,filename$_2$><,filename>

where o =

T - List the table of contents for the program file named in <filename$_1$> . Additional program files may be specified in <filename$_2$>

F - Lists directory information about a file

No option - List the symbolic element. <filename$_1$> must specify an element name. Any number of symbolic elements may be listed by this option.

E.3.5 The @MOVE Command

The @MOVE command will move any tape file forward or backward over a specified number of end-of-file marks.

The @MOVE statement has the following format:

@MOVE, <o>ƀ<filename$_1$>,n

where o =

If the 'option' field is omitted the tape is moved forward. If the tape is to be moved backward, the option is "B".

filename - name of the tape file to be positioned

n - number (decimal) of end-of-file marks past which the tape is to be moved.

E.3.6 The @REWIND Command

The @REWIND command rewinds a magnetic tape file to the load point. The format of this command is:

@REWIND,[I]b̸<filename$_1$><,filename$_2$>...

where

I - specifies interlock, i.e., the peripheral unit may not be accessed after rewind until the operator has cleared the interlock condition.

<filename$_1$> - specifies the names of the files to be rewound.

CR - carriage return

b̸ - blank

< > - item inside carats to be provided by user

> - Univac 1110 prompt character

NNNN - underlined item to be entered by user

NNNN - non-underlined item response by computer

Figure E-1

Notation Conventions

APPENDIX F GLOSSARY

Acoustical Survey	A program of measurements taken throughout a community to determine noise levels within the community.
Alternate Print File	A user specified data file that is used in place of the temporary system created PRINT$ file to save the print generated by a run so that it can be looked at later. The alternate file is set up by the statement @'BRKPT PRINT$/file-name', and terminated by an '@BRKPT PRINT$' statement.
Archive	To store data or a file in some safe, permanent manner, usually on tape.
Case	The data obtained from one respondent. Four records are used to hold the data for each case.
Compiler	A compiler is a program that prepares a machine language program (all ones and zeros) from a high level language program (such as FORTRAN, COBOL, PL/I).
Disk	A memory device consisting of swiftly spinning platter coated with an easily magnetized substance, which is read from or written to by a moving recording head. Also called 'mass-storage device' because of its large capacity.
Editor	A program used to create and modify program files and data files, at the direction of the user.
Element	A UNIVAC term which indicates a named portion of a program file.
EXEC 8	The brand name of the operating system used on the UNIVAC 1100 series computer. The NCC EXEC 8 version follows general UNIVAC philosophy, but should be considered unique to that site.

Field	An arbitrarily defined portion of a data record reserved for a particular use.
File	A collection of related records treated as a unit. For UNIVAC there are two distinct types: program files and data files. Data files differ from program files in that they contain no elements.
Flag	A field set to indicate one of two possible conditions. Within a program, a variable used for this purpose.
Hardware	All or part of the machinery of which a computer system is composed. This includes the computers (CPUs), core memory, drum and disk storage units, magnetic tape drives, multiplexers, modems, telephones, terminals, card readers, card punches, line printers, etc.
Interpreter	A program which translates source language into machine code and executes it line by line.
Keys	Under EXEC 8, keys are passwords used to gain access to a file. Keys may be used to control read or write access or both. EXEC 8 considers keys to be part of the file name. They must be specified when @ASGing the file: @ASG,U qualifier*filename/read-key/write-key.
LISTEN	Local Information System to Evaluate Noise. A data gathering and analysis system designed to determine the impact of noise on a community.
Operating Systems	A collection of software that directs the action of the computer in much of the routine 'housekeeping' tasks. The typical operating system includes an executive, or supervisor routine, an input/output control routine, utility programs for file handling, an assembler, and one or more compilers.
Priority	An operating system assigned attribute of a run which determines when it will be processed. Priority depends on a number of factors including user-requested priority, resources necessary for the run (core, time, tape drives, etc.), and time of day.

Question	In this manual, refers to a question asked by the attitudinal survey.
Quiet Communities Program	An EPA project intended to show communities how to apply the best available techniques to combat noise.
Recode	To change all occurences of a given value in a survey question to some other value. For example, question 1 of the attitudinal survey is recoded by substituting '0.5' for all occurences of '95'.
Remote Site	A site, located outside of the computer center, where batch jobs may be submitted and their output received. It consists of at least a card reader and line printer; often a card punch is added.
Respondent	A person who has answered a questionnaire.
Run	A sequence of Executive Control Language statements which is submitted to, and processed by, EXEC8. The first statement of a run is always the 'RUN' statement and the last statement is always the '@FIN' statement. Collectively, the group is often referred to as a 'runstream'.
Runstream	See RUN
Qualifier	An additional 12 characters prefixed to a file name to insure uniqueness. When omitted, the project id from the @RUN statement is used. The qualifier and file name are always separated with an asterisk '*'.
Submit	Entering a runstream into the system for processing by employing a card reader or terminal.
System Processor	A utility program which operates on data.
Systems Programmer	A programmer whose specialty is computer operating systems.
Utility Program	Any useful program maintained and supplied by the computer center, for the benefit of the user.

Variable	In this manual, data from the attitudinal survey that is used in the SPSS data analysis. Simple variables are variables in which the data comes from only one survey question (e.g., the variable RTHQ9). Compound variables are variables in which the data from a number of survey variables are combined (e.g., the variable PUBSERV).

ATTITUDINAL SURVEY DATA ANALYSIS SYSTEM (ASDAS)

MAINTENANCE MANUAL

Prepared for
OFFICE OF NOISE ABATEMENT
AND CONTROL
U.S. ENVIRONMENTAL PROTECTION AGENCY

CONTRACT NO. 68-01-3840

SEPTEMBER 29, 1979

COMPUTER SCIENCES CORPORATION

6565 Arlington Boulevard
Falls Church, Virginia 22046

Major Offices and Facilities Throughout the World

159<

TABLE OF CONTENTS

Page Number

SECTION 1 - INTRODUCTION

1.1 Summary Overview.......................... 1-1
1.1 History.................................... 1-1
1.3 Environment................................ 1-2

SECTION 2 - SYSTEM OVERVIEW

2.1 Software Functions......................... 2-1
2.2 Program Hierachy........................... 2-1
2.3 Inputs..................................... 2-1
2.4 Processing................................. 2-3
2.5 Outputs.................................... 2-3
2.6 Interfaces................................. 2-4

SECTION 3 - PROGRAM DESCRIPTIONS

3.1 The CLEAN Program.......................... 3-1
3.1.1 Software Function........................ 3-1
3.1.2 Inputs................................... 3-1
3.1.3 Processing............................... 3-1
3.1.4 Output................................... 3-15
3.1.5 Interface Requirements................... 3-15
3.1.6 Data Description......................... 3-15
3.2 The VARIANCE Program....................... 3-17
3.2.1 Software Function........................ 3-17
3.2.2 Inputs................................... 3-17
3.2.3 Processing............................... 3-17
3.2.3.1 Program Flow........................... 3-17
3.2.3.1.1 Subroutine RECODE.................... 3-17
3.2.3.1.2 Subroutine MISSNG.................... 3-20
3.2.3.1.3 Subroutine RANKR..................... 3-20
3.2.3.1.4 Subroutine SORT...................... 3-20
3.2.3.1.5 Subroutine STAT...................... 3-22
3.2.3.2 Constraints/Limitations................ 3-24
3.2.3.3 Execution Time Requirements............ 3-24
3.2.3.4 Peripheral Storage Requirements........ 3-24
3.2.4 VARIANCE Outputs......................... 3-24
3.2.5 Interface Requirements 3-24
3.2.6 Program Variable Description............. 3-24
3.2.7 Program Execution Requirements........... 3-29
3.3 The TAILOR Program......................... 3-29
3.3.1 Software Function........................ 3-29
3.3.2 Inputs................................... 3-29
3.3.3 Processing............................... 3-29
3.3.4 Outputs.................................. 3-34
3.3.5 Interface Requirements................... 3-34

Page Number

3.3.6 Data Description.......... 3-34
3.4 The RECODE1 Program.......... 3-35
3.4.1 Software Function.......... 3-35
3.4.2 Inputs.......... 3-35
3.4.3 Processing.......... 3-36
3.4.3.1 Program Flow.......... 3-36
3.4.3.1.1 SPSS Initiation Statements.......... 3-36
3.4.3.1.2 Variable Creation and Modification.......... 3-36
3.4.3.1.3 User Inserted Operations.......... 3-38
3.4.3.1.4 Labeling.......... 3-38
3.4.3.1.5 Missing Value Definition.......... 3-39
3.4.3.1.6 SPSS Parameter Definitions.......... 3-39
3.4.3.1.7 Z-score Production.......... 3-39
3.4.3.1.8 System File Creation.......... 3-40
3.4.3.2 Constraints/Limitations.......... 3-40
3.4.3.3 Execution Time Requirements.......... 3-40
3.4.3.4 Peripheral Storage Requirements.......... 3-40
3.4.3.5 Differences between Forms of RECODE1.......... 3-40
3.4.4 RECODE1 Outputs.......... 3-41
3.4.5 Interface Requirements.......... 3-41
3.4.6 Program Variable Description.......... 3-43
3.4.7 Program Execution Requirements.......... 3-43
3.5 The RECODE2 Program.......... 3-45
3.5.1 Software Function.......... 3-45
3.5.2 Inputs.......... 3-45
3.5.3 Processing.......... 3-45
3.5.3.1 Program Flow.......... 3-46
3.5.3.2 Constraints/Limitations.......... 3-46
3.5.3.3 Execution Time Requirements.......... 3-49
3.5.3.4 Peripheral Storage Requirements.......... 3-49
3.5.3.5 Differences between forms of RECODE2.......... 3-49
3.5.4 Outputs.......... 3-50
3.5.5 Interface Requirements.......... 3-51
3.5.6 Program Variable Description.......... 3-51
3.5.7 Program Execution Requirements.......... 3-51
3.6 The FREQUENCIES, CROSSTABS, AND REGRESSIONS Programs.......... 3-52
3.6.1 Software Functions.......... 3-52
3.6.2 Processing.......... 3-53
3.6.2.1 Program Flow.......... 3-53
3.6.2.2 Constraints/Limitations.......... 3-54
3.6.2.3 Execution Time Requirements.......... 3-54
3.6.2.4 Peripheral Storage Requirements.......... 3-54
3.6.2.5 Differences between Forms of FREQUENCIES CROSSTABS and REGRESSIONS.......... 3-55
3.6.3 Outputs.......... 3-56
3.6.4 Interface Requirements.......... 3-56
3.6.5 Data Description.......... 3-57
3.6.6 Program Execution Requirements.......... 3-57

TABLE OF FIGURES

Figure Number	Name	Page
2-1	System Flow Chart	2-2
2-2	ASDAS Relationship Within LISTEN	2-5
3-1	CLEAN Flow Chart	3-10
3-2	Variable for which VARIANCE Calculates Statistics	3-18
3-3	VARIANCE Flow Chart	3-19
3-4	Changes made by the RECODE subroutine in VARIANCE	3-21
3-5	STAT Flow Chart	3-23
3-6	TAILOR Flow Chart	3-30
3-7	RECODE1 Flow Chart	3-37
3-8	Variables for which RECODE1 Calculates Scores	3-42
3-9	Mnemonically Named Variables Created in RECODE1	3-44
3-10	RECODE2 Flow Chart	3-47

LIST OF TABLES

Table Number	Name	Page
3-1	The CLEAN Initialization Card	3-2
3-2	Questionnaire Data Cards: Layouts and Legal Values	3-3
3-3	Variables and Common Blocks in VARIANCE	3-25
3-4	Variables Created in RECODE2	3-48

1.1 Summary Overview

This manual describes the Attitudinal Survey Data Analysis System (ASDAS). ASDAS processes data from the Community Noise Assessment Social Survey, a survey of community attitudes toward noise. Its ultimate product is a statistical analysis of the data, including:

- frequency tables
- crosstabulation tables
- and, a series of multiple linear regression fits.

Intermediate outputs are:

- clean (verified) survey data
- a statistical analysis to indicate the reliability of the survey's measurements.

1.2 History

The ASDAS reflects the social survey questionnaire as of October 1, 1979. It is a completely new system reflecting the major changes made in the questionnaire after its test in Allentown, PA in 1978.

The original questionnaire, survey methodology and data analysis computer programs were developed by Wyle Laboratories for the State and Local Program's division of the EPA's Office of Noise Abatement and Control (ONAC). Wyle also developed the new questionnaire. The data analysis system for the new questionnaire, the ASDAS, was designed and programmed by Computer Sciences Corporation. The new system has greater

- flexibility and
- maintainability

than had the old. The increased flexibility permits the use of slightly varying questionnaires to accommodate community differences. It also enables the users to perform only those portions of the statistical analysis which are useful to him.

Maintainability is achieved through

- Readability
- Documentation
- Modularization.

All are much improved over the original system.

1.3 Environment

Program Sponsor: Environmental Protection Agency
Office of Noise Abatement and Control
State and Local Programs Division

Project Officer: Leonard Libster

Software Developer: Computer Sciences Corporation
Applied Technologies Division

Programmers: Michael Glazer, task leader
Dr. Raghu Singh
Steven Eagleson
Kim Nguyen

Implemented on: UNIVAC 1110

At: EPA National Computer Center (NCC)
Research Triangle Park, North Carolina

Operating System: EXEC 8 as modified at NCC

System software used: SPSS Version H Release 7.2-UW 1.0 Sept 1978
FORTRAN V (V4R1)
MAP (29R1)

Anticipated Users: EPA contractors, probably data entry personnel

Section 2 - System Overview

2.1 Software Functions

The system performs three primary functions:

- Detection of errors in the data
- Variance analysis
- Statistical analysis

It also provides a subsidiary capability:

- a way to select the statistical analysis procedures appropriate to the community.

Other capabilities are outlined in the Users' Manual, but these are incidental to the system.

2.2 Program Hierarchy

The programs comprising the system and their functions are listed below:

Program	Function
CLEAN	Error detection
VARIANCE	Variance Analysis
TAILOR	Statistical Analysis Selection
RECODE1	
RECODE2	
FREQUENCIES	Statistical Analysis
CROSSTABS	
REGRESSIONS	

A system flow chart is shown in Figure 2-1.

2.3 Inputs

Inputs to the system are:

Program	Input
CLEAN	Raw questionnaire data Flag card
TAILOR	SPSS template program Flag card New code (optional, Accoustical Data

In addition, the user must also correct the questionnaire data.

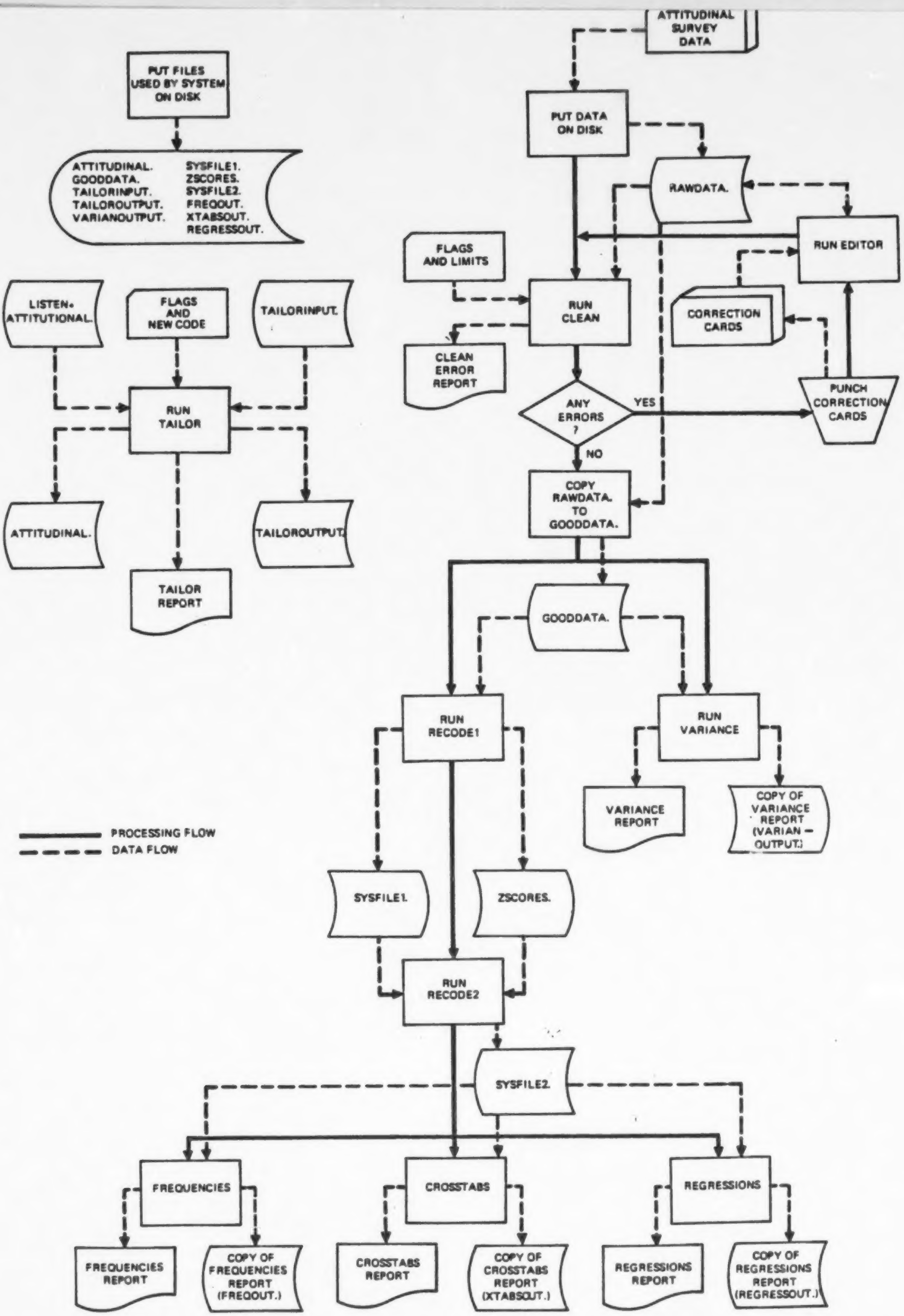

PUT FILES USED BY SYSTEM ON DISK
ATTITUDINAL.
GOODDATA.
TAILORINPUT.
TAILOROUTPUT.
VARIANOUTPUT.
SYSFILE1.
ZSCORES.
SYSFILE2.
FREQOUT.
XTABSOUT.
REGRESSOUT.
LISTEN. ATTITUTIONAL.
FLAGS AND NEW CODE
TAILORINPUT.
RUN TAILOR
ATTITUDINAL.
TAILOR REPORT
TAILOROUTPUT.
ATTITUDINAL SURVEY DATA
PUT DATA ON DISK
RAWDATA.
RUN EDITOR
FLAGS AND LIMITS
RUN CLEAN
CORRECTION CARDS
CLEAN ERROR REPORT
ANY ERRORS ?
YES
NO
PUNCH CORRECTION CARDS
COPY RAWDATA. TO GOODDATA.
GOODDATA.
RUN RECODE1
RUN VARIANCE
VARIANCE REPORT
COPY OF VARIANCE REPORT (VARIAN — OUTPUT.)
PROCESSING FLOW
DATA FLOW
SYSFILE1.
ZSCORES.
RUN RECODE2
SYSFILE2.
FREQUENCIES
CROSSTABS
REGRESSIONS
FREQUENCIES REPORT
COPY OF FREQUENCIES REPORT (FREQOUT.)
CROSSTABS REPORT
COPY OF CROSSTABS REPORT (XTABSOUT.)
REGRESSIONS REPORT
COPY OF REGRESSIONS REPORT (REGRESSOUT.)

Figure 2-1

System Flow Chart

167<

2.4 Processing

The system's first step is to analyze the questionnaire's raw data for errors. This is done by the CLEAN program which produces a listing of the errors it detects. The inputs to CLEAN are shown in subsection 2.3.

The user must correct the errors that CLEAN has revealed. Then CLEAN is again run on the data. This process continues until all errors have been eliminated. The resulting, correct, data is used as input to the VARIANCE program. The statistics produced by this program are written to the default line printer.

At some point before the SPSS statistical analysis programs are used, the TAILOR program must be run to produce the correct form of these programs. Inputs to TAILOR are described in the previous subsection. "New code" refers to acoustical survey data which may be available and any other SPSS statements the user may wish to insert in the programs.

After they have been transformed into the appropriate configurations, the SPSS programs are run on the corrected data. The first program used is RECODE1 which takes the data, performs some manipulations on them, and then outputs them to disk in SPSS system file format. RECODE1 also produces Z-scores, or standardized scores, for a number of variables. These are also written to disk, but in a different format and to a different file.

RECODE2, the next program, receives the system file and Z-scores produced by RECODE1 as input. It performs its own manipulations, producing a single SPSS system file as output.

Finally, the FREQUENCIES, CROSSTABS and REGRESSIONS programs are run using the file output by RECODE2 as input. These three programs are totally independent of one another. Each produces a printed report.

2.5 Outputs

Outputs from the system are:

Program	Output
CLEAN process	Corrected data
VARIANCE	Printed report
RECODE2	SPSS system file containing data

168<

Program	Output
FREQUENCIES	Printed report
CROSSTABS	Printed report
REGRESSIONS	Printed report

2.6 Interfaces

Figure 2-2 shows ASDAS' place in the LISTEN project.

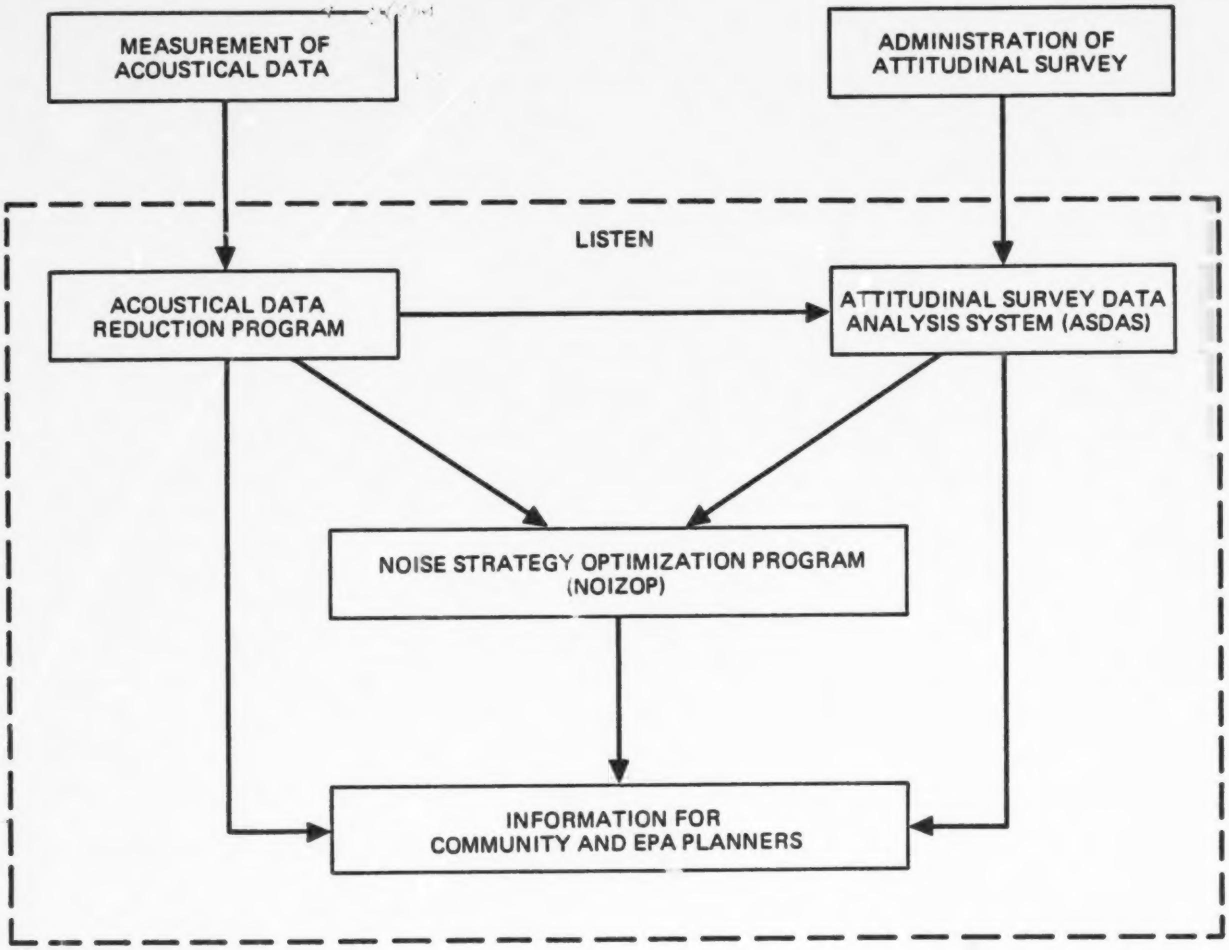

MEASUREMENT OF ACOUSTICAL DATA
ADMINISTRATION OF ATTITUDINAL SURVEY
LISTEN
ACOUSTICAL DATA REDUCTION PROGRAM
ATTITUDINAL SURVEY DATA ANALYSIS SYSTEM (ASDAS)
NOISE STRATEGY OPTIMIZATION PROGRAM (NOIZOP)
INFORMATION FOR COMMUNITY AND EPA PLANNERS

Figure 2-2

ASDAS Relationship to Listen

Section 3 - Program Descriptions

3.1 The CLEAN Program

This subsection describes CLEAN V1.0(0). The executable absolute file element and FORTRAN V source language file elements are found in LISTEN* ATTITUDINAL.CLEAN file.

3.1.1 Software Function

CLEAN has only one function. This is to range check each individual item of data in all the attitudinal survey cases presented to it. Any deviation from expected values or inconsistency between items cause a descriptive error message to be generated.

3.1.2 Inputs

CLEAN receives input from two sources:

- The job runstream containing the CLEAN initialization card
- The file proj-id*RAWDATA., containing uncleaned attitudinal survey data

The CLEAN initialization card format is described by Table 3-1 and in the ASDAS User Manual, subsection 3.2.2. The runstream is the source only by default. An @USE 5,file-name.statement will cause the specified file to be read for the initialization card.

The file proj-id*RAWDATA. is created by the ASDAS system initiation procedure. It is an SDF file with 80 character card images punched according to the attitudinal survey format (see Reference 1, Appendix A). The data from each questionnaire is coded onto four cards. This is called a case. Table 3-2 lists the card data fields and their legal values.

3.1.3 Processing

The major functions and logic flow of the main program are shown in Figure 3-1.

The most important functions of the main program are to:

- Initialize the flexible data ranges
- Set the CTLPRG switch and MAXERR

Table 3-1

THE CLEAN INITIALIZATION CARD

FIELD NAME	COLS	STANDARD VALUE	REMARKS
LOID	1-5	00001	
	6	blank	
HIID	7-11	01500	
LONZ	12-16	00001	
HINZ	17-21	00017	
LOAR	22-26	00001	
HIAR	27-31	00007	
LOCLUS	32-36	00001	
HICLUS	37-41	00100	
CTY1	42-46		If less than four city codes are used, set the extra fields to one of the codes that is used
CTY2	47-51		
CTY3	52-56		
CTY 4	57-61		
MAX INTERVIEW DATE	62-66		Community dependent
MAXERR	67-71	00009	
	72	blank	
CTLPRGS	73		1=without 2=with (a community noise control program)
	74-80	blank	

Table 3-2

Questionnaire Data Cards:

Layouts and Legal Values

CARD ONE

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID NO.*	1-5	00001	01500		Cannot = previous CARD 1 ID #
Card	6-7			01	
Noise Zone*	8-9	01	16		
Area*	10	1	7		
Cluster*	11-13	001	100		
City*	14-16			Community Dependent	
Selection Label	17-18	00	12		
Number of Calls	19	1	9		
Interview Date*	20-21	01	Community Dependent		
Total Number of Adult Residents	22	1	8		
Respondent Number on List	23	1	8		Must be < or = to TOT. NO. ADULT RESIDENTS
Sex of Respondent	24			1, 2	Must correspond to CARD 4, B5
Age of Respondent	25-26	18	99		Must correspond to CARD 4, Q22
	27-80				Blanks Only

*With flexible data ranges set at recommended values.

CARD TWO

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID No.	1-5	00001	01500		Must match CARD 1 ID
Card	6-7			02	
Q1	8-9	01	90	95, 98, 99	
Q2	10	1	5	8, 9	
Q3 (A-C)	11-12	00	32	98, 99	
	13-14				
	15-16				
Q4 (A-C)	17-19	101	111	0, 199	
	20-22	201	214	299	
	23-25	301	310	399	
		401	404	499	
		501	506	599	
		601	603	699	
		701	704	799	
		801	807	899	
		901	906	950, 975 998, 999	
Q5 (A-H)	26,27	1	5	8,9	
	28,29				
	30,31				
	32,33				

Continued Next Page

CARD TWO Continued

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
Q6(A-K)	34,35	1	5	8,9	
	36,37				
	38,39				
	40,41				
	42,43				
	44				
Q7	45	1	5	8,9	
Q8	46	1	4	8,9	
Q9	47	1	9		
Q10(A-D)	48,49	1	5	8,9	
	50,51				
Q11(A-F)	52,53			0,1,8,9	
	54,55				
	56,57				
Q12	58	1	3	8,9	
Q13(A-E)	59,60	1	5	8,9	
	61,62				
	63				
Q14	64	1	5	8,9	
Q15(A-G)	65,66			0,1,8,9	
	67,68				
	69,70				

Continued Next Page

CARD TWO Continued

FIELD NAME	Cols.	Min.	Mac.	Fixed Val.	Additional Constraints
Q15(A-G) (Cont.)	71				
Q16(A-C)	72,73			0,1,8,9	
	74				
	75-80				Blanks Only

CARD THREE

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID No.	1-5	00001	01500		Must = CARD 1 ID NO.
Card	6-7			03	
Q17(A-T)	8,9,10	1	9		
	11,12				
	13,14				
	15,16				
	17,18				
	19,20				
	21,22				
	23,24				
	25,26				
	27				
Q17(U-V)	28,29	00	99		
	30,31				
Q18(A-F)	32,33			0,1,8,9*	
	34,35			8+	
	36,37				
Q19	38			1,2,8,9*+	
Q19A(A)	39	0	2	8,9*	If zero then Q19 must = 2. If 1,2 8, or 9 then Q19 must = 1
		1	4+		
Q19A(B-D)	40,41	0	2	8,9*	11
	42			8+	Must = 8 if 'with' is indicated

*Without Noise Control Program (CLTPRG=1)
+With Noise Control Program (CTLPRG=2)

Continued Next Page

CARD THREE Continued

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
Q20	43	1	9		
Q21(A-H)	44,45			0,1,8,9	
	46,47				
	48,49				
	50,51				
	52-80				Blanks Only

CARD FOUR

FIELD NAME	Cols.	Min.	Max.	Fixed Val.	Additional Constraints
ID NO.	1-5	00001	01500		Must = CARD 1 ID #
Card	6-7			04	
Q22	8-9	18	99		
Q23	10-11	00	21	98,99	
Q24A	12-13	01	07	80,98,99	
Q25A	14-15	00	15	98,99	Must = 0 if Q24A = 80, Must not = 0 if Q24A Not = 80
Q26	16-17	01	10	98,99	
Q27	18	1	9		
Q28	19	1	4	8,9	
Q29	20-21	01	99		
Time	22-23	00	99		
B1	24			1,2,9	
B2	25	0	9		
B3	26-28	000	999		
B4	29			1,2,9	
B5	30			1,2,9	
B6	31	1	3	9	
	32-80				Blanks Only

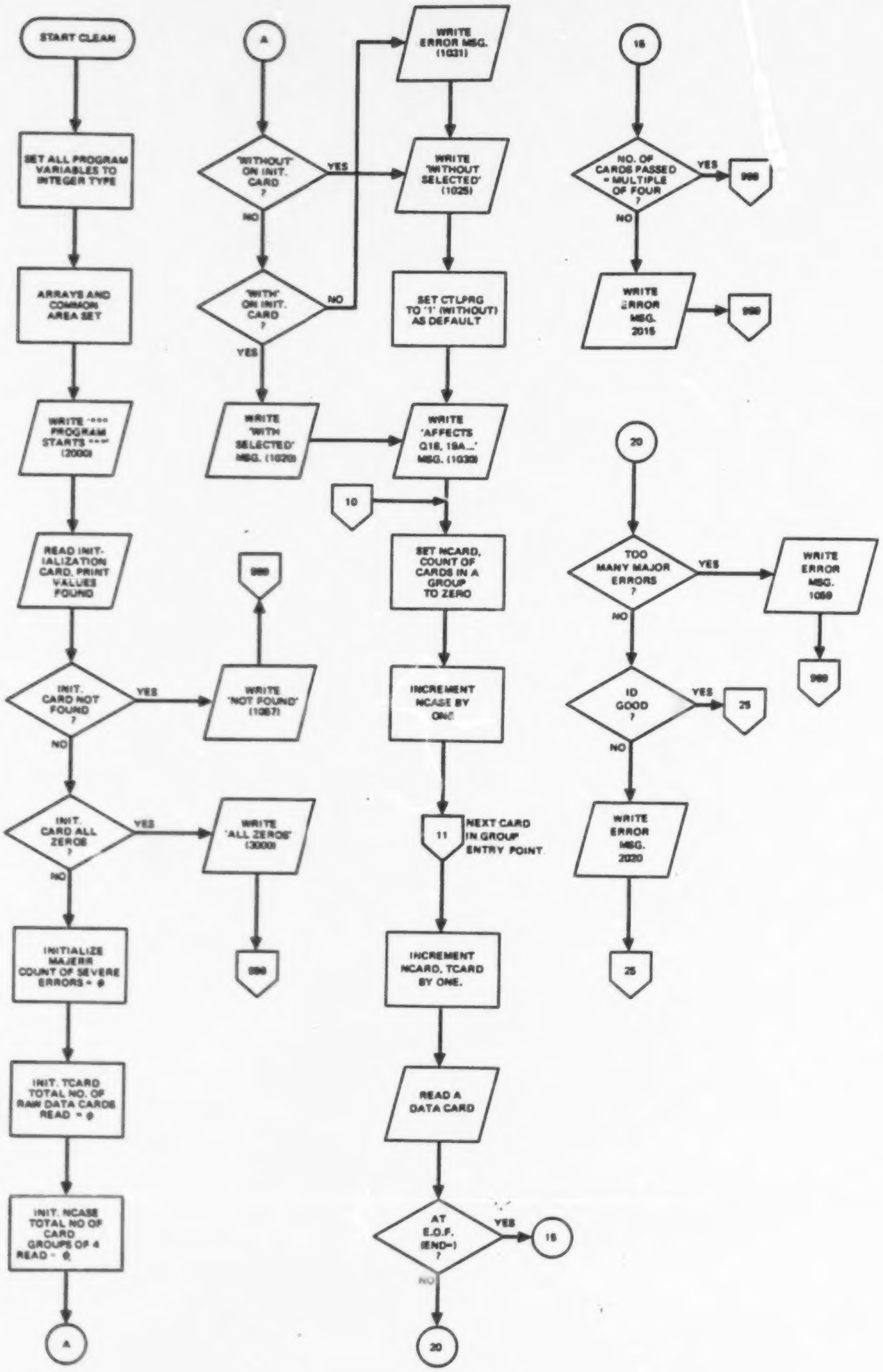

START CLEAN
SET ALL PROGRAM VARIABLES TO INTEGER TYPE
ARRAYS AND COMMON AREA SET
WRITE "*** PROGRAM STARTS ***" (2000)
READ INITIALIZATION CARD, PRINT VALUES FOUND
INIT. CARD NOT FOUND ?
YES
WRITE 'NOT FOUND' (1057)
NO
INIT. CARD ALL ZEROS ?
YES
WRITE 'ALL ZEROS' (3000)
NO
INITIALIZE MAJERR COUNT OF SEVERE ERRORS = 0
INIT. TCARD TOTAL NO. OF RAW DATA CARDS READ = 0
INIT. NCASE TOTAL NO OF CARD GROUPS OF 4 READ = 0
A
A
'WITHOUT' ON INIT. CARD ?
YES
NO
'WITH' ON INIT. CARD ?
NO
YES
WRITE 'WITH SELECTED' MSG. (1020)
WRITE ERROR MSG. (1031)
WRITE 'WITHOUT SELECTED' (1025)
SET CTLPRG TO '1' (WITHOUT) AS DEFAULT
WRITE 'AFFECTS Q18, 19A...' MSG. (1030)
10
SET NCARD, COUNT OF CARDS IN A GROUP TO ZERO
INCREMENT NCASE BY ONE
11
NEXT CARD IN GROUP ENTRY POINT
INCREMENT NCARD, TCARD BY ONE.
READ A DATA CARD
AT E.O.F. (END=) ?
YES
15
NO
20
15
NO. OF CARDS PASSED = MULTIPLE OF FOUR ?
YES
NO
WRITE ERROR MSG. 2015
20
TOO MANY MAJOR ERRORS ?
YES
WRITE ERROR MSG. 1050
NO
ID GOOD ?
YES
25
NO
WRITE ERROR MSG. 2020
25

Figure 3-1a

CLEAN Flow Chart

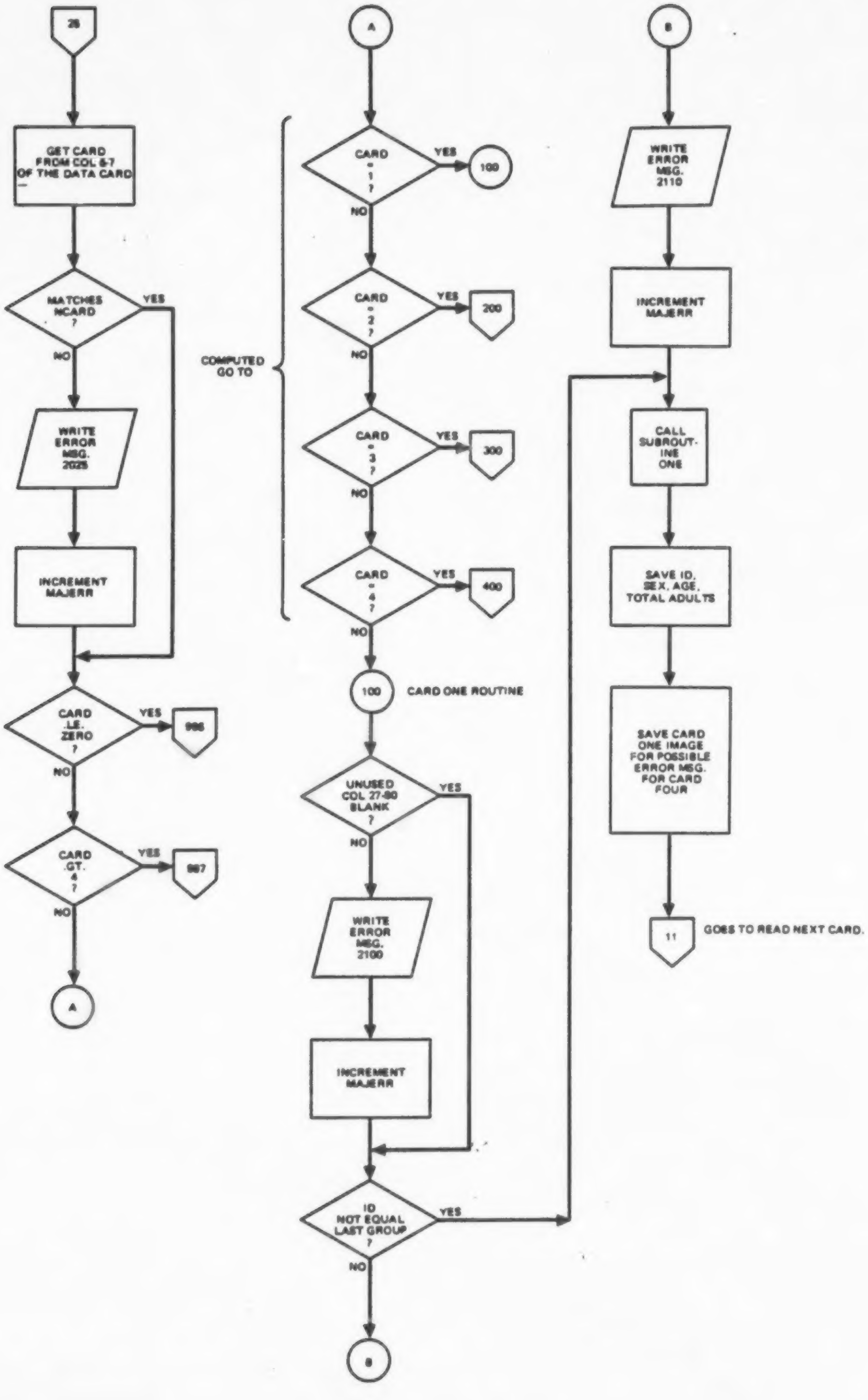

25
GET CARD FROM COL 6-7 OF THE DATA CARD
MATCHES NCARD ?
YES
NO
WRITE ERROR MSG. 2025
INCREMENT MAJERR
CARD .LE. ZERO ?
YES
996
NO
CARD .GT. 4 ?
YES
997
NO
A
A
COMPUTED GO TO
CARD = 1 ?
YES
100
NO
CARD = 2 ?
YES
200
NO
CARD = 3 ?
YES
300
NO
CARD = 4 ?
YES
400
NO
100
CARD ONE ROUTINE
UNUSED COL 27-80 BLANK ?
YES
NO
WRITE ERROR MSG. 2100
INCREMENT MAJERR
ID NOT EQUAL LAST GROUP ?
YES
NO
B
B
WRITE ERROR MSG. 2110
INCREMENT MAJERR
CALL SUBROUT-INE ONE
SAVE ID, SEX, AGE, TOTAL ADULTS
SAVE CARD ONE IMAGE FOR POSSIBLE ERROR MSG. FOR CARD FOUR
11
GOES TO READ NEXT CARD.

Figure 3-1b

CLEAN Flow Chart

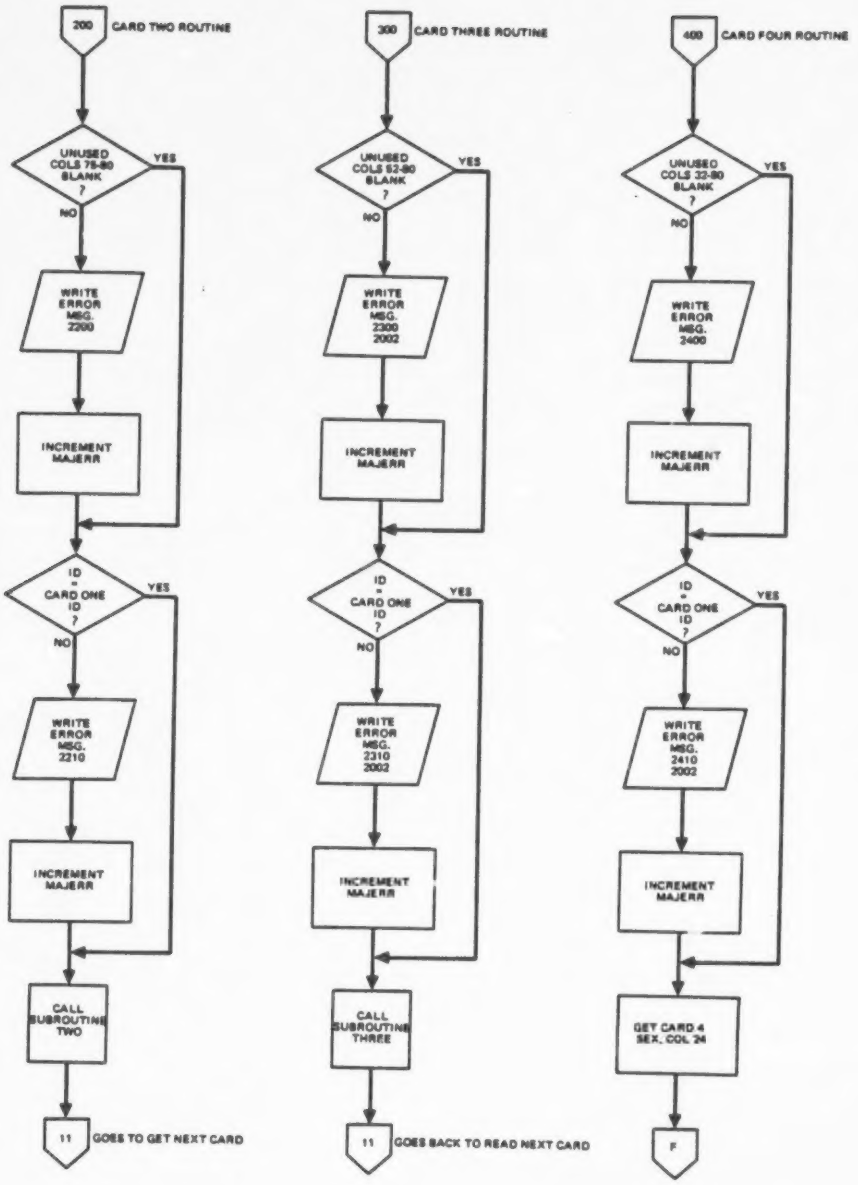

200
CARD TWO ROUTINE
UNUSED COLS 75-80 BLANK ?
YES
NO
WRITE ERROR MSG. 2200
INCREMENT MAJERR
ID = CARD ONE ID ?
YES
NO
WRITE ERROR MSG. 2210
INCREMENT MAJERR
CALL SUBROUTINE TWO
11
GOES TO GET NEXT CARD
300
CARD THREE ROUTINE
UNUSED COLS 52-80 BLANK ?
YES
NO
WRITE ERROR MSG. 2300 2002
INCREMENT MAJERR
ID = CARD ONE ID ?
YES
NO
WRITE ERROR MSG. 2310 2002
INCREMENT MAJERR
CALL SUBROUTINE THREE
11
GOES BACK TO READ NEXT CARD
400
CARD FOUR ROUTINE
UNUSED COLS 32-80 BLANK ?
YES
NO
WRITE ERROR MSG. 2400
INCREMENT MAJERR
ID = CARD ONE ID ?
YES
NO
WRITE ERROR MSG. 2410 2002
INCREMENT MAJERR
GET CARD 4 SEX, COL 24
F

Figure 3-1c
CLEAN Flow Chart

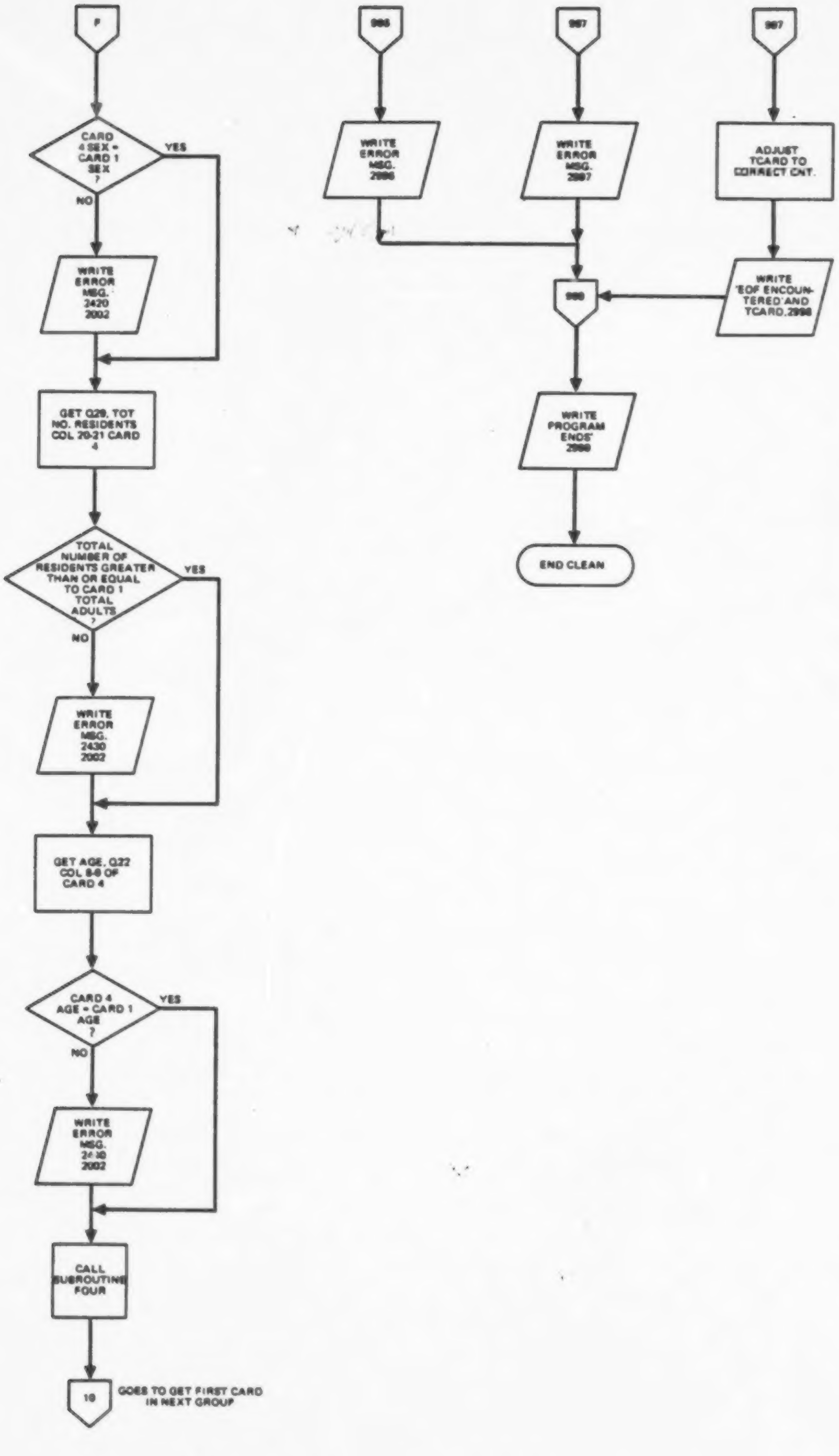

P
CARD 4 SEX = CARD 1 SEX ?
YES
NO
WRITE ERROR MSG. 2420 2002
GET Q29, TOT NO. RESIDENTS COL 20-21 CARD 4
TOTAL NUMBER OF RESIDENTS GREATER THAN OR EQUAL TO CARD 1 TOTAL ADULTS ?
YES
NO
WRITE ERROR MSG. 2430 2002
GET AGE, Q22 COL 8-9 OF CARD 4
CARD 4 AGE = CARD 1 AGE ?
YES
NO
WRITE ERROR MSG. 2440 2002
CALL SUBROUTINE FOUR
10
GOES TO GET FIRST CARD IN NEXT GROUP
998
WRITE ERROR MSG. 2998
997
WRITE ERROR MSG. 2997
997
ADJUST TCARD TO CORRECT CNT.
WRITE 'EOF ENCOUNTERED' AND TCARD,2998
999
WRITE 'PROGRAM ENDS' 2999
END CLEAN

Figure 3-1d

CLEAN Flow-Chart

- Read the questionnaire data
- Check the count of major errors against MAXERR
- Check fields common to all cards
- Branch to program segments dealing with specific cards.

Flexible data ranges are provided to allow for expected variations in survey administration. Thus, while one community may have completed the interview process within 30 days, another may have taken 60. Instead of setting the value for the maximum legal interview day to some arbitrary number (such as 60), CLEAN allows the exact day count to be specified. This provides more thorough data checking.

CTLPRG and MAXERR are similar to the flexible data ranges in that they increase CLEAN's versatility. CTLPRG allows two different questionnaire formats to be used; one for communities with noise control programs and one for communities without such programs. MAXERR is the maximum number of major errors (basically those errors detected in the main program) which must occur before CLEAN terminates prematurely.

The program segments that deal with the individual cards have these operations in common:

- Unused column check
- Identification number check
- A call to a subroutine to check field ranges.

In addition, the segment for card one saves data from some fields to check against card four data. The segment for card four checks data against card one.

The four subroutines (ONE, TWO, THREE, FOUR) each check the data fields on one of the four cards of a case. The subroutine number corresponds to the card checked. The logic flow within each subroutine is just a series of IF-statements branching around error message WRITES. Because of its simplicity, they will not be flow charted.

COLS is a function which converts a multi-column data field into an INTEGER value.

3.1.4 Output

All CLEAN output is to the job's PRINT$ file. This is the default that is selected when the FORTRAN unit number 6 is written to the job runstream without an @USE 6,file-name statement, for FORTRAN V Programs.

CLEAN will always produce a banner indicating the values of the initialization card and a '*** PROGRAM ENDS ***' line to show that data cleaning has occurred. All errors found will be indicated by messages between the two. If '*** PROGRAM ENDS ***' is not displayed, the program has not run properly.

3.1.5 Interface Requirements

Before CLEAN can be used, the raw data must be put on disk. CLEAN is designed to run standalone, perhaps many times, before the next step in the ASDAS system is begun. It is suggested that a good file editor be kept handy (such as the University of Maryland @US*ER.ED processor) to make the corrections directly to proj-id*RAWDATA.

3.1.6 Data Description

COL(80) is in COMMON block STOR0, and is the record input buffer. Each element receives 1 column of a data card. It is common to all subroutines.

COL1(80) is used by CLEAN MAIN to save a card image of the most recent card one data card. If any cross-consistency errors are found in the other data cards, COL1 is used in the error message.

ID =	is given the data card's ID, cols 1-5
LIST(4)	is in COMMON block STOR0, and is EQUIVALENCEd to 4 major variables in the program. LIST is used to write the value in error messages. LIST(1)=TCARD, LIST(2)=NCASE, LIST(3)=ID, LIST(4)=NCARD
MAJERR	Count of severe errors in data cards
NCARD	is the count of records in a case
NCASE	is the count of cases
TCARD	is the total count of records read

ALPHA	An array in COMMON area ALPH, passes character date for some error messages, (so that Q19AA can be printed instead of Q19A(1))
NCPC	Number of cards per case, set in DATA statement to 4 value.
CTLPRG	Switch in COMMON block LIMITS, set by col 73 of initialization card, a 1 means Q19A (card 3) is checked for 1,2,8,9 if Q19 equals 1, and 0 if Q19 equals 2. A 2 for CTLPRG means that Q19A part A can only equal 1,2,3,4 regardless of the value of Q19, and Q19AB-Q19AD must equal 8. These values reflect changes to the survey depending on whether the community has a noise control problem.
INITCK	A counter incremented after the initialization card is read. If it is zero then the READS statement, END=BRANCH was directed around the increment statement, indicating that a data card did not follow the @XQT statement in the runstream.
PID	Holds card 1 ID for a case, all other cards in case must match.

3.2 The VARIANCE Program

This subsection describes VARIANCE V1.0(0). It is a FORTRAN program.

3.2.1 Software Function

VARIANCE calculates a number of statistics for certain attitudinal survey questions. The statistics calculated are:

Mean
Mean squared error
Standard error
Confidence interval
Coefficient of variation

The variables for which statistics are calculated are listed in Figure 3-2.

3.2.2 Inputs

Input to VARIANCE are:

- Cleaned questionnaire data
- The measure of size (MOS) for the city involved.

Cleaned questionnaire data is read from FORTRAN unit 10. The data are in the format described in Section 3.1.2 above. If the system is operated according to the Users' Manual's instructions, the data will be in a file named proj-id*GOODDATA.

The MOS is read from FORTRAN unit 5. It is a single number in I5 format left justified on the card. It is the total number of households that exist in the community in which the data were gathered.

3.2.3 Processing

3.2.3.1 Program Flow

Figure 3-3 shows a high-level flow chart for the VARIANCE program. Each major entry on the chart is described separately in the subsections below.

3.2.3.1.1 Subroutine RECODE

The RECODE subroutine changes the values of certain questions. This is necessary to make the statistics for these questions meaningful. For example,

Q1
Q2
Q5A through Q5H
Q6A through Q6K
Q7
Q8
Q9
Q10A through Q10D
Q12
Q13A through Q13E
Q16A through Q16C
Q17A through Q17T
Q19
Q20
Q22
Q25A
Q28
Q29

Figure 3-2

Variables for which VARIANCE Calculates Statistics

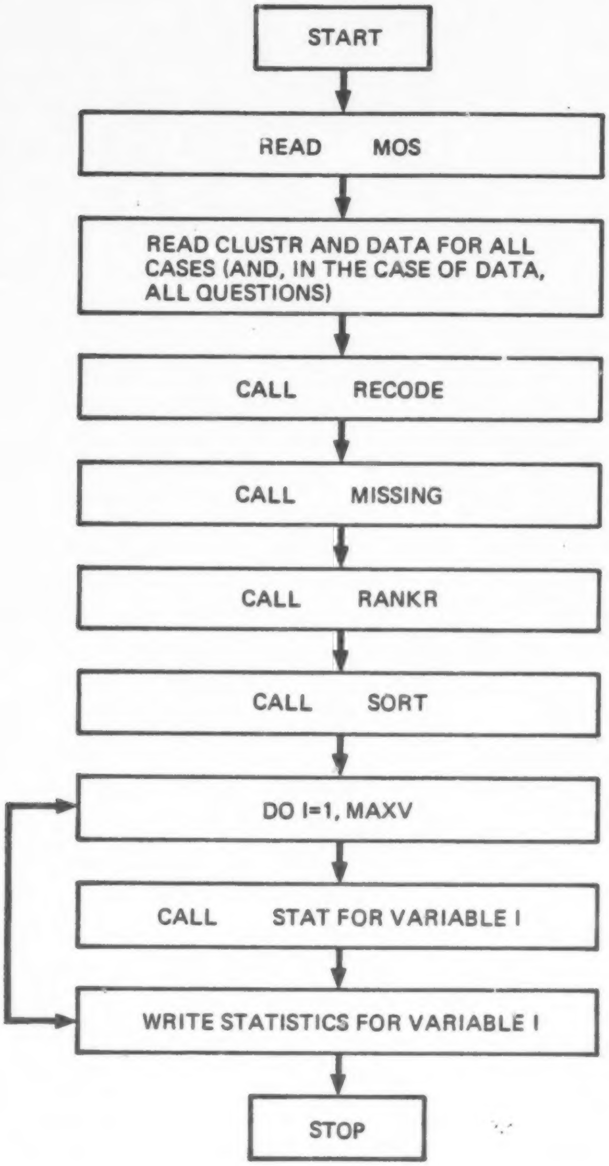

START
READ MOS
READ CLUSTR AND DATA FOR ALL CASES (AND, IN THE CASE OF DATA, ALL QUESTIONS)
CALL RECODE
CALL MISSING
CALL RANKR
CALL SORT
DO I=1, MAXV
CALL STAT FOR VARIABLE I
WRITE STATISTICS FOR VARIABLE I
STOP

Figure 3-3

VARIANCE Flow Chart

in question 1, the values 90 and 95 have special meanings which do not fit with the rest of this question's value codes. They are changed to appropriate values. Figure 3-4 shows the questions altered as well as the old values and the recoded values for these questions.

3.2.3.1.2 Subroutine MISSNG

The MISSNG subroutine replaces those data values which indicate missing data (respondent couldn't or wouldn't answer the question) with a flag value (-999.0).

Missing values are indicated in the original data by 8 and 9 for questions coded in one card column, or 98 and 99 for questions coded in two card columns.

3.2.3.1.3 Subroutine RANKR

RANKR sorts the elements of the array CLSRNK. CLSRNK's elements are pointers into the array CLUSTR which contains the cluster numbers for each case in the order in which the cases were read by VARIANCE. After RANKR is finished, CLSRNK is ordered such that the value of its first element is the position in the CLUSTR array of the lowest cluster number. Each succeeding element of CLSRNK then points to an equal or larger cluster number in CLUSTR. This is necessary to permit subroutine SORT to operate correctly. RANKR operates by:

- Assuming an initial ranking
- Performing a bubble sort to make this ranking correct

3.2.3.1.4 Subroutine SORT

SORT takes the array CLSRNK as ordered by RANKR and uses its elements as indexes into the array CLUSTR to create a third array, RDRDCL, which contains an ordered list of the cluster numbers that occur in the data (least to greatest). Each cluster number appears once in RDRDCL.

SORT also counts the number of cluster numbers found and stores this number in NCLFND. It may have occurred to the reader that SORT is something of a misnomer for this subroutine. This is correct.

RDRDCL is necessary for the pairing of clusters and calculations of statistics in STAT.

QUESTION	OLD VALUE	MEANING	RECODED VALUE
Q1	90	All my life	Set equal to Q22 (age)
	95	Six months or less	0.5
Q20	1	$0.10	0.1
	2	0.25	0.25
	3	0.50	0.5
	4	1.00	1.0
	5	2.50	2.5
	6	More than $2.50	2.75
	7	None	0.00
		Thurstone Annoyance Scale	
Q9, Q17A to Q17T	1	Tremendously Annoyed	6.8
	2	Highly Annoyed	6.1
	3	Considerably Annoyed	4.9
	4	Medium Annoyed	3.8
	5	Partially Annoyed	2.9
	6	A Little Annoyed	2.2
	7	Not at all Annoyed	1.1

Figure 3.4

Changes made by the RECODE subroutine in VARIANCE

3.2.3.1.5 Subroutine STAT

The STAT subroutine calculates the statistics that the VARIANCE program produces. STAT also counts the number of data points in the sample. The statistics are calculated according to the following formulas:

$r = \text{mean} = y/x$

$$\text{MSE} = \text{mean squared error } (r) = \frac{1-f}{X2} \sum_{i} ((y_i - rz_i) - (y_{i+1} - rx_{i+1}))$$

where $i = 1, 3, 5 \ldots n$,

$r = \frac{700}{\text{MOS}}$ (where MOS=total number of households in the surveyed community)

x = total number of non-missing data points

y=sum of all non-missing data values

x_i=total number of non-missing data points for the ith cluster

y_i=sum of all non-missing data values for the ith cluster

SE=standard error= MSE

The confidence interval is:

High: R+1.96.(SE)

Low: R-1.96. (SE)

CV=coefficient of variation = $\frac{SE}{r}$.

Figure 3-5 shows a flow chart for STAT. STAT first makes a list of the clusters containing all missing data and uses them as the first entries in a list of clusters to be ignored in the calculation of statistics (SKPDCL). If there is an odd number of clusters, the cluster with the highest cluster number is also placed in that list (provided it wasn't already there).

Each of the above clusters is then paired with a second cluster according to the following rule: if the cluster number is odd, it is paired with the cluster following it in the ordered list of cluster numbers (RDRDCL); if the cluster number is even, it is paired with the cluster preceding it. The new members of the pairs are added to the list of clusters to be ignored (SKPDCL). All cases which have cluster numbers listed in SKPDCL now have the data for the question being analyzed set to -1. At this point, STAT constructs a list of clusters that will be used in the data analysis (USEDCL). These clusters are paired according to their position in that list and the

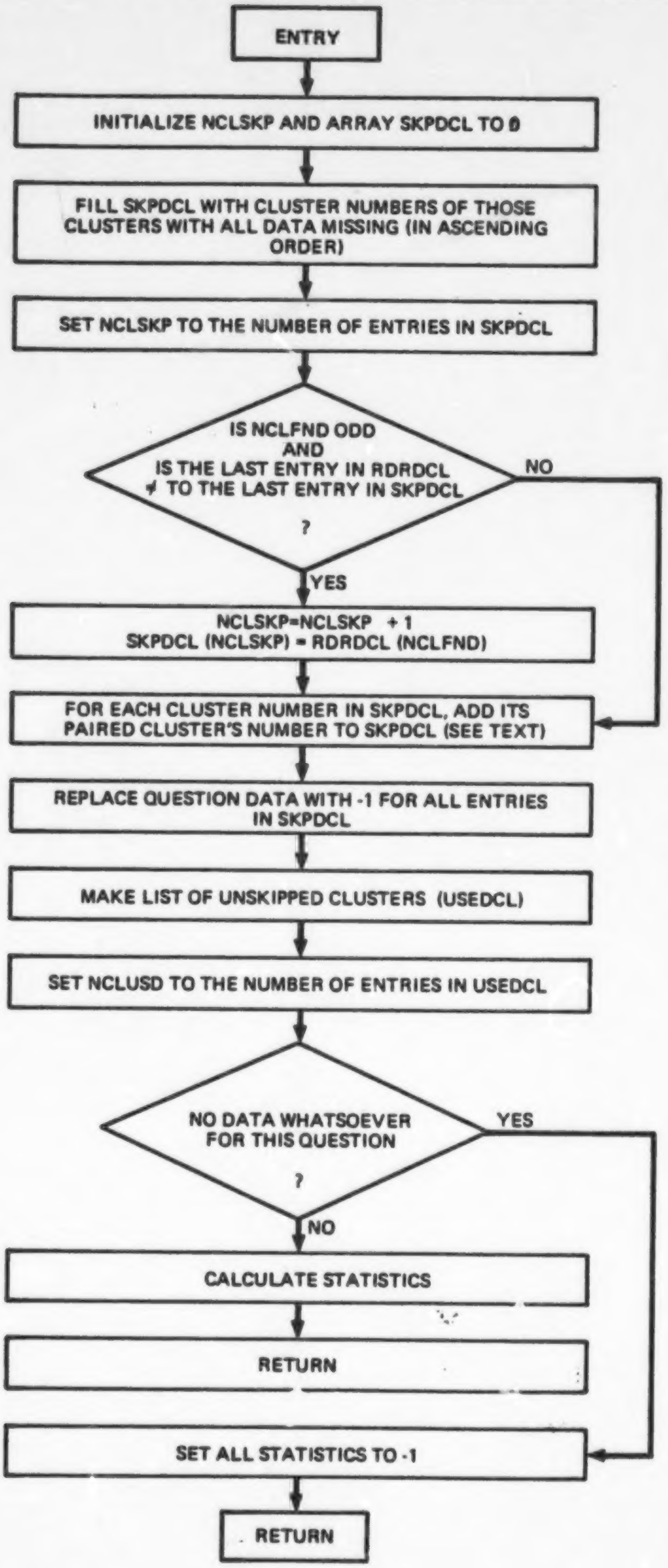

ENTRY
INITIALIZE NCLSKP AND ARRAY SKPDCL TO 0
FILL SKPDCL WITH CLUSTER NUMBERS OF THOSE CLUSTERS WITH ALL DATA MISSING (IN ASCENDING ORDER)
SET NCLSKP TO THE NUMBER OF ENTRIES IN SKPDCL
IS NCLFND ODD AND IS THE LAST ENTRY IN RDRDCL ≠ TO THE LAST ENTRY IN SKPDCL ?
NO
YES
NCLSKP=NCLSKP + 1
SKPDCL (NCLSKP) = RDRDCL (NCLFND)
FOR EACH CLUSTER NUMBER IN SKPDCL, ADD ITS PAIRED CLUSTER'S NUMBER TO SKPDCL (SEE TEXT)
REPLACE QUESTION DATA WITH -1 FOR ALL ENTRIES IN SKPDCL
MAKE LIST OF UNSKIPPED CLUSTERS (USEDCL)
SET NCLUSD TO THE NUMBER OF ENTRIES IN USEDCL
NO DATA WHATSOEVER FOR THIS QUESTION ?
YES
NO
CALCULATE STATISTICS
RETURN
SET ALL STATISTICS TO -1
RETURN

Figure 3-5

STAT Flow Chart

statistics calculated as shown in the formulas above. If a question has no data, all statistics are set to -1.0 except, of course, for the number of points, which is zero.

3.2.3.2 Constraints/Limitations

VARIANCE is unable to distinguish between good and bad data. Therefore, the data cleaning step must be performed correctly.

VARIANCE is also limited as to the number of cases of data it can process. It will take data from up to 760 respondents. The program will fail in an unpredictable manner if this limit is exceeded.

3.2.3.3 Execution Time Requirements

To process 700 cases of input data, VARIANCE requires 64 kWords of main memory and 50 SUPS.

3.2.3.4 Peripheral Storage Requirements

VARIANCE requires approximately:

- o 5 tracks of disk space for its output file.

3.2.4 VARIANCE Outputs

VARIANCE writes its output to FORTRAN unit 6, the default line printer. It consists of about 20 pages of statistics calculated by the program. If the ASDAS is run as described in the Users' Manual, this output will appear in two forms:

- A printout
- A disk file, in EXEC 8 PRINT$ format, which contains a copy of the above printout

3.2.5 Interface Requirements

Before running VARIANCE it is necessary that the data cleaning process has been performed correctly and completely, including the copying of the file proj-id*RAWDATA. into the file proj-id*GOODDATA.

3.2.6 Program Variable Description

Table 3-3 lists the variables used in VARIANCE and describes them.

Table 3-3

Variance and Common Blocks in Variance

GLOBAL VARIABLES

LABEL	PROGRAM SEGMENT IN WHICH USED	CONTENTS AND PURPOSE	TYPE/ARRAY SIZE
CLSRNK	RANKR, SORT, STAT	Array containing indexes into CLUSTR	INTEGER/(760)
CLUSTR	MAIN, RANKR, SORT STAT	Array containing cluster numbers as read from data cards	INTEGER/(760)
COEFVR	MAIN, STAT	Coefficient of variation as computed by STAT	INTEGER
CONFHI	MAIN, STAT	High end of confidence interval	REAL
CONFLO	MAIN, STAT	Low end of confidence interval	REAL
DATA	MAIN, MISSING, RECODE, STAT	Array containing all data from data cards except cluster number	REAL/(64,760)
IDATA	MISSING, RECODE	Integer version of particular items in DATA	INTEGER
MAXV	MAIN, MISSNG	Number of variables for which statistics are to be computed	INTEGER
MEAN	MAIN, STAT	Mean value statistic; returned by STAT	REAL
MOS	MAIN, STAT	Measure of size (number of households in the community)	INTEGER
NCASE	MAIN, MISSING RANKR, RECODE, STAT	Number of cases of questionnaire data read	INTEGER
NCLFND	SORT, STAT	Number of distinct cluster numbers in data; number of entries in RDRDCL	INTEGER

BEL	PROGRAM SEGMENT IN WHICH USED	CONTENTS AND PURPOSE	TYPE/ARRAY SIZE
CLUSD	MAIN, STAT	Number of entries in USEDCL	INTEGER
NTS	MAIN, STAT	Number of data points used in the calculation of the statistics	REAL
DRDCL	SORT, STAT	List of distinct cluster numbers found in questionnaire data; arranged in numerical order	INTEGER/(300)
GMA	MAIN, STAT	Standard error	REAL
ARNUM	MAIN, MISSNG STAT	Row number (first index) of DATA array which corresponds to question whose statistics are to be calculated	INTEGER
ARORD	MAIN, MISSNG	Array containing Row number (first index) of DATA array for each variable for which statistics will be calculated, in the order in which the calculations are to be performed	INTEGER/(64)
RIANS	MAIN, STAT	Mean Square Error	REAL
		LOCAL VARIABLES	
LSTR	MAIN	Type real version of NCLSTR; used for output	REAL
LUMN	RECODE	Do loop index	INTEGER
LRVAL	RECODE	Array containing dollar equivalents of questionnaire codes for question 20	REAL/(7)
NCTR	MAIN	Counter indicating total number of lines output to this point	INTEGER
ME	MAIN	Array containing labels for questions whose statistics are calculated in Hollerith format	INTEGER/(64)

ABEL	PROGRAM SEGMENT IN WHICH USED	CONTENTS AND PURPOSE	TYPE/ARRAY SIZE
3OX	MISSNG	Array containing the number of card columns occupied by the question for each question analyzed	INTEGER/(64)
CLSKP	STAT	Number of entries in SKPDCL	INTEGER
EWPAG	MAIN	Array containing points (in terms of total lines of VARIANCE output) at which a new page should be started	INTEGER/(12)
KPDCL	STAT	Array containing the cluster numbers of those clusters whose data will be ignored in calculating statistics	INTEGER/(300)
UM	STAT	Value of the summation involving y, r, and x shown in the text	REAL
HRSTV	RECODE	Array containing Thurstone annoyance scale values for questionnaire codes in Q9 and Q17A through Q17T	REAL/(7)
USEDCL	STAT	Array containing cluster numbers of those clusters whose data will be used in the calculation of statistics	INTEGER/(300)
ALMIS	MISSNG	Contains value which replaces missing data indicators in questionnaire data	REAL
	STAT	Total number of data points	INTEGER
H		Array containing the number of valid data points in each cluster (ordered like USEDCL)	REAL/(100)
	STAT	Sum of all data values	REAL
YH	STAT	Sum of all data values occurring in a cluster (ordered like USEDCL)	REAL/(100)

These variables are used as DO loop indexes, DO loop limits and temporary storage locations:

I, IB, II, IJ, IS, ITEMP, JJ, K, KK, N, NN, NOT, NU

		COMMON BLOCKS
BLOCK1	ALL	Contains DATA array
BLOCK2	ALL	Contains NCASE and MAXV
LST	SORT, STAT	Contains RDRDCL array and NCLFND
ORDER	MAIN, MISSING RANKR, SORT, STAT	Contains CLUSTR and CLSRNK and VARORD arrays
SIZE	MAIN, STAT	Contains MOS

3.2.7 Program Execution Requirements

The input file (proj-id*GOODDATA. is the suggested filename) must be assigned. It must be @USEd to the correct internal name: 10. If it is desired to save the program's output on disk, an alternate print file must be set up. The name suggested for this file is proj-id*VARIANOUTPUT.

The card containing the MOS data must be inserted directly after the @XQT LISTEN*ATTITUDINAL.VARIANCE card.

3.3 The TAILOR Program

This sub-section describes TAILOR V1.0(0) program. The executable absolute element and FORTRAN V source language elements are found in LISTEN*ATTITUDINAL.TAILOR.

3.3.1 Software Function

Different versions of the SPSS programs are used depending upon whether or not a community has a noise control program and whether or not it has performed an acoustical survey. TAILOR constructs SPSS programs appropriate to a community.

TAILOR also provides a facility that allows the user to add additional SPSS statements at certain points in these programs.

3.3.2 Inputs

As input, TAILOR expects a flag setting record from FORTRAN unit 5. The default action is to look in the runstream file following the @XQT. The program Users' Manual describes the format, flag names, and purpose of this record. Any SPSS records to be inserted should follow the flag record. They should be ended with an @EOF statement.

The other source of input is FORTRAN unit number 4. This is the SPSS template source (see below).

3.3.3 Processing

TAILOR's first action is to read a card containing two flags (see flowchart, Figure 3-6). The first of these flags, ACUSFL, is set to a non-zero value to indicate that an acoustical survey has been performed in this community. NSPFLG is set to a non-zero value to indicate that a noise control program is active in this community.

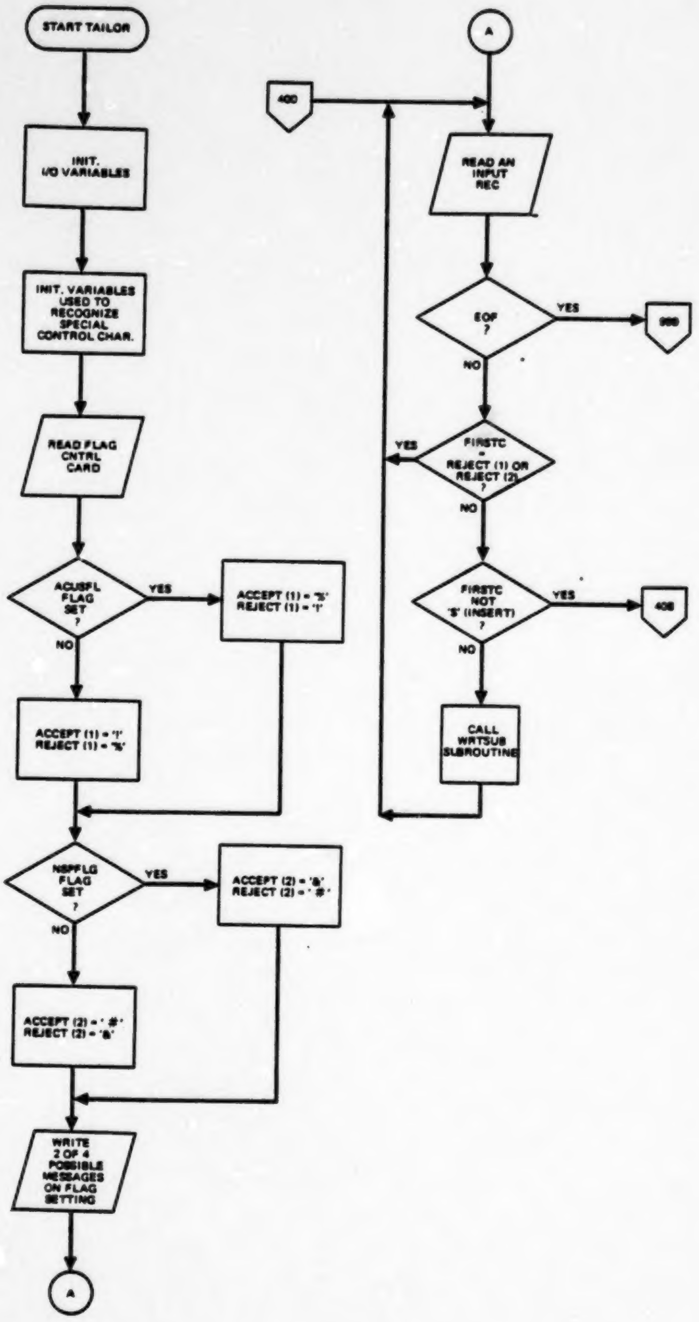

START TAILOR
INIT.
I/O VARIABLES
INIT. VARIABLES
USED TO
RECOGNIZE
SPECIAL
CONTROL CHAR.
READ FLAG
CNTRL
CARD
ACUSFL
FLAG
SET
?
YES
NO
ACCEPT (1) = '%'
REJECT (1) = '!'
ACCEPT (1) = '!'
REJECT (1) = '%'
NSPFLG
FLAG
SET
?
YES
NO
ACCEPT (2) = '&'
REJECT (2) = ' #'
ACCEPT (2) = ' #'
REJECT (2) = '&'
WRITE
2 OF 4
POSSIBLE
MESSAGES
ON FLAG
SETTING
A
A
400
READ AN
INPUT
REC
EOF
?
YES
999
NO
FIRSTC
=
REJECT (1) OR
REJECT (2)
?
YES
NO
FIRSTC
NOT
'$' (INSERT)
?
YES
400
NO
CALL
WRTSUB
SUBROUTINE

Figure 3-6a

TAILOR Flow Chart

200<

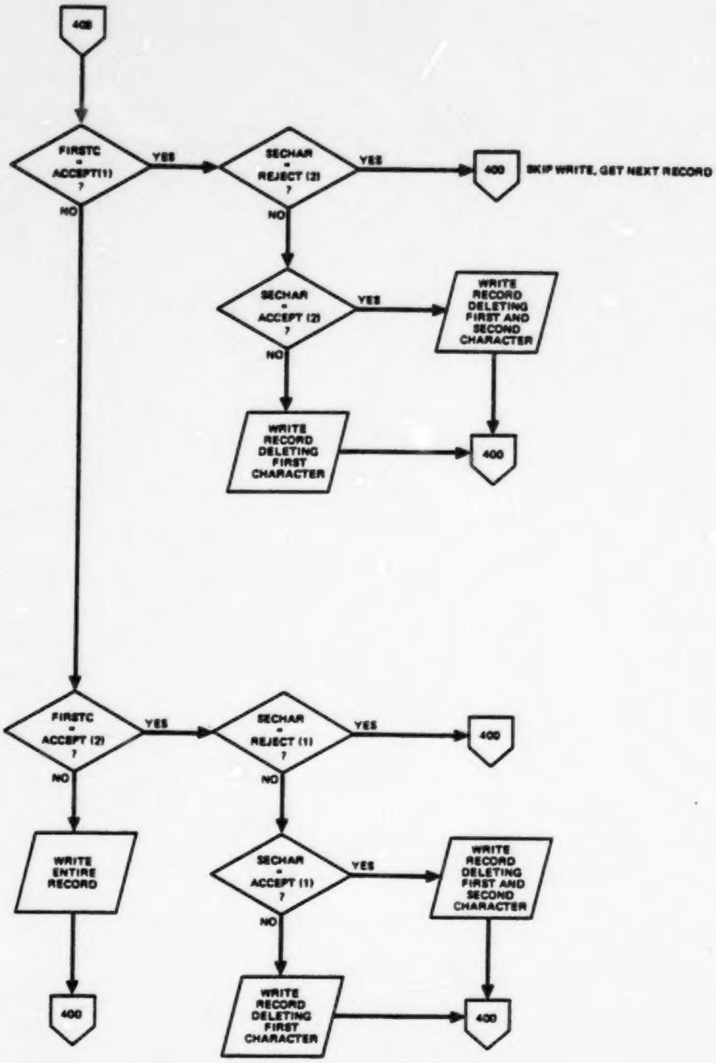

408
FIRSTC = ACCEPT(1) ?
YES
NO
SECHAR = REJECT (2) ?
YES
400
SKIP WRITE, GET NEXT RECORD
NO
SECHAR = ACCEPT (2) ?
YES
WRITE RECORD DELETING FIRST AND SECOND CHARACTER
NO
WRITE RECORD DELETING FIRST CHARACTER
400
FIRSTC = ACCEPT (2) ?
YES
NO
SECHAR = REJECT (1) ?
YES
400
NO
WRITE ENTIRE RECORD
400
SECHAR = ACCEPT (1) ?
YES
WRITE RECORD DELETING FIRST AND SECOND CHARACTER
NO
WRITE RECORD DELETING FIRST CHARACTER
400

Figure 3-6b

TAILOR Flow Chart

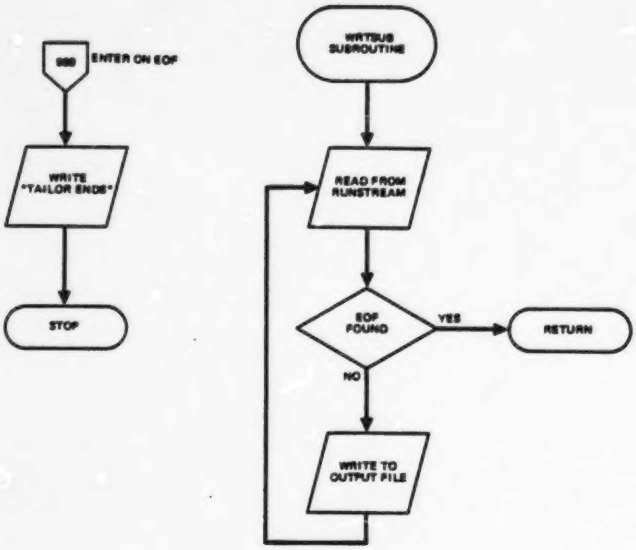

999
ENTER ON EOF
WRITE "TAILOR ENDS"
STOP
WRTSUB SUBROUTINE
READ FROM RUNSTREAM
EOF FOUND
YES
RETURN
NO
WRITE TO OUTPUT FILE

Figure 3-6c

TAILOR Flow Chart

TAILOR next reads, record by record, a 'template' of the SPSS program to be generated. A template is simply a standard SPSS program in which certain statements are preceded by one or two control characters. These characters indicate to TAILOR under what conditions the statements are to be included in the final SPSS program.

The control characters and actions taken by TAILOR are:

%	Marked template statements are written to the output file only when ACUSFL flag is set, and are skipped over (deleted) when it is not. (i.e., the statement is only for communities with an acoustical monitoring program.)
!	Marked template statements are skipped over only when ACUSFL flag is set, and are written to the output file when it is not. (i.e., the statement is only for communities with no acoustical monitoring program.)
&	Marked template statements are written to the output file only when NSPFLG flag is set, and are skipped over if it is not set. (i.e., the statement is only for communities with a noise control program.)
#	Marked template statements are skipped over (deleted) when NSPFLG flag is set, and are written to the output file when it is not set. (i.e., the statement is only for communities without a noise control program.)
$	(dollar sign) marks the point in the file where insertion is to take place from the remaining records of unit 5 file. The $ marked record is always skipped over (deleted).

The logic of the program cannot deal with the control character combinations &# (#&) and %! (!%) in the first two columns of a template record, since these pairings are, of course, contradictory. All other possibilities are legal.

The TAILOR program continues until the template file is exhausted.

Note that array equivalency is used to shift the marked column(s) to the left, out of the record, as it is sent to the output file.

3.3.4 OUTPUTS

TAILOR writes the SPSS scource language statements to FORTRAN unit number 7 after processing. Also, in the runstream printout, two message lines are printed that indicate the flag conditions selected.

3.3.5 INTERFACE REQUIREMENTS

None

3.3.6 DATA DESCRIPTION

ACUSFL	is a flag field in columns 1-5 of the TAILOR control card. By making this a non-zero, non-blank, value the user indicates that the surveyed community has an acoustical monitoring program. Zero or blank indicates that the community has no acoustical monitoring program. The monitoring program is relevant to the attitudinal survey since it can be used to supply real noise level data to the SPSS programs in place of the values otherwise used in the program.
NSPFLG	is a flag field in columns 6-10 of the TAILOR control card. It is used to indicate whether or not the surveyed community has a noise control program. Zero or blank means there is no such program and 1 or greater tells TAILOR that there is a noise control program. The existence of a noise program is important because it means that a modified version of the survey (changes to Q18, !19, Q19A, Q21) should have been used.
LINE(32)	is an array that receives the input records, where LINE(1) and LINE(2) are given the first two columns
OUTLIN1(31)	is an array equivalenced to part of the LINE array, and is used to output the records that have 1 special control character in column array equivalencing
OUTLIN2(30)	is an array equivalenced to part of the LINE array, and is used to output the records that have 2 special control characters in columns 1 and 2

The Statement

```
EQUIVALENCE (LINE(3),OUTLN1(2),OUTLN2(1))
```

causes an apparent shift to the left that will eliminate the special control characters from the record.

3.4 THE RECODE1 PROGRAM

This subsection describes RECODE1 V1.0(0). The template for this program is found in the element LISTEN*ATTITUDINAL.RECODE1. (See subsection 3.3 above for an explanation of the function of a template). RECODE1 is written in SPSS.

3.4.1 SOFTWARE FUNCTION

RECODE1 has six functions:

- Modify the cleaned survey data for use by the FREQUENCIES, CROSSTABS and REGRESSIONS programs
- Label all questions and those data values for which a label is appropriate
- Identify to SPSS those values that indicate missing date for each survey question
- Perform certain transformations on the data to reduce core memory and time requirement in the FREQUENCIES and CROSSTABS programs
- Store all data, original and modified, labels and other information in an SPSS system file. This file will be labeled for SPSS as SYSFILE1
- Calculate standardized scores, or z-scores, for a number of survey questions and write these z-scores to desk

Each of these functions is described in subsection 3.4.3

3.4.2 INPUTS

RECODE1 receives input from one source:

- proj-id*GOODDATA., a file containing cleaned attitudinal survey data

The file must be an SDF (System Data Format) file. The data is arranged in 80 character card images. Four card images are used per respondent. This is called a "case". Table 3-2 shows the location of each question on the cards of a case.

3.4.3 PROCESSING

The RECODE1 used by a given city is actually one of two very similar forms, of the program. The TAILOR program, described in 3.3 above, generates the correct RECODE1 program to use on a city's data. Differences between the two RECODE1 forms are noted in Section 3.4.3.5 below.

3.4.3.1 Program Flow

Program flow is linear since there is no branching in SPSS.

See Figure 3-7 for a flow chart of RECODE1.

3.4.3.1.1 SPSS Initiation Statements

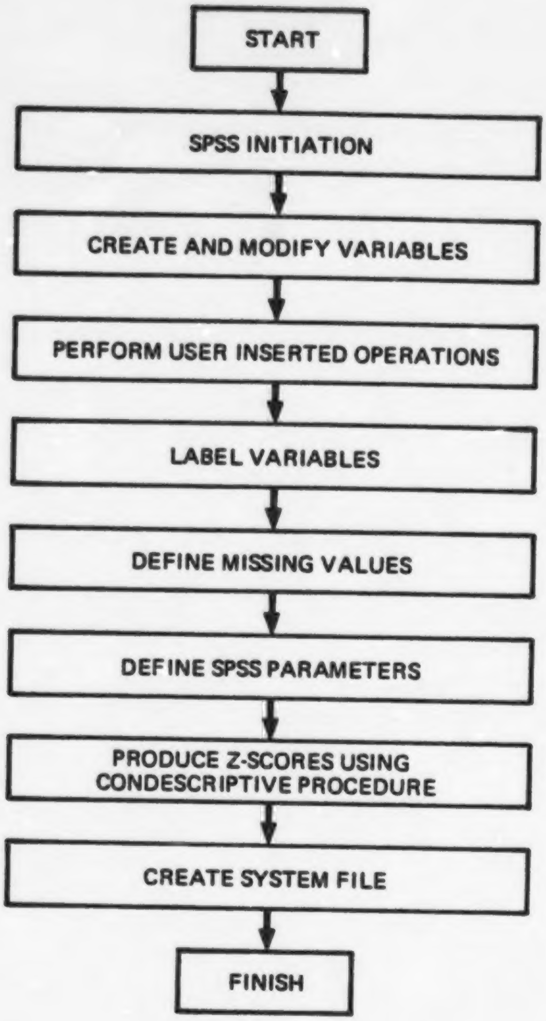
The initiation statements:

- Allocate memory for the program (see pp. 113 and 618, Reference 6, Appendix A)
- Name the run (p. 72, ibid)
- Give the system file the name SYSFILE1 (p. 35, ibid)
- Define the location of each question's data on the card image constituting a case (p. 49, ibid)
- Define the input unit as TAPE, i.e., unit 8 (pp. 41 and 614, ibid)

3.4.3.1.2 Variable Creation and Modification

Variables are created and modified for the following reasons:

- The FREQUENCIES, CROSSTABS or REGRESSIONS programs cannot produce meaningful output from the original variable.
- The RECODE2 program needs a modified variable to calculate a composite variable (see subsection 3.5)
- To save core memory and time during execution



START
SPSS INITIATION
CREATE AND MODIFY VARIABLES
PERFORM USER INSERTED OPERATIONS
LABEL VARIABLES
DEFINE MISSING VALUES
DEFINE SPSS PARAMETERS
PRODUCE Z-SCORES USING CONDESCRIPTIVE PROCEDURE
CREATE SYSTEM FILE
FINISH

Figure 3-7

RECODE1 Flow Chart

An example of a modification performed for the first reason is that done on question 1, length of residence. The values for this question are, with the exception of several special values, equal to the respondent's length of residence in years. 95 is one of the special values. It indicates a length of residence of six months or less. It is recoded to 0.5 (representing 0.5 years) to enable the FREQUENCIES, CROSSTABS and REGRESSIONS programs to operate correctly.

RECODE2 uses modified forms of some variables to compute compound variables. The way in which the calculations are done makes it necessary to change the values that indicate missing data for these questions.

Core memory and time are saved by limiting the range of a values a variable may assume. For certain procedures, SPSS sets up and accesses arrays. These arrays have as many locations as the variable has assumable values. If a variable assumes the values 1, 2, 8 and 9 then recoding the values 8 and 9 to -1 allows FREQUENCIES and CROSSTABS to use less than half the memory than would otherwise be required. Time otherwise spent accessing this memory is also saved. The legitimacy of such a recode is shown in 3.4.3.1.5 below.

The SPSS instructions used to create the new, modified variables are: COMPUTE, RECODE, COUNT, DO REPEAT and END REPEAT (see pp. 89-101, 107-110, 121-126 reference 6, Appendix A). A new variable is created each time an original variable is modified. This preserves the original data.

3.4.3.1.3 User Inserted Operations

The instructions the user has inserted by means of the TAILOR program are performed at this point.

3.4.3.1.4 Labeling

Variables and variable values are labeled to make reading the tables produced by the FREQUENCIES, CROSSTABS and REGRESSIONS programs easier. Labels are defined by the SPSS instructions VAR LABELS and VALUE LABELS (see pp. 59 and 62, Reference 6, Appendix A).

The labeling commands are placed in a different element than RECODE1. This is to allow the TAILOR program to be run on both elements independently. The element in which the labels reside is LISTEN*ATTITUDINAL.LABELS.

Since TAILOR is run on this element, there are four slightly differing versions of this segment of RECODE1. The differences are noted in 3.4,3.5 below. The segment is inserted in RECODE1 by the @ADD ATTITUDINAL.LABELS statement in RECODE1.

The variable labels were taken from section 5.2.1 of reference 1, Appendix A. Value labels were adapted from the questionnaire itself.

3.4.3.1.5 Missing Value Definitions

For many of the survey's questions it is possible for the respondent to either refuse to answer or not know what answer to give. When this occurs, the question is said to have missing data. On the card input, missing data is represented by special codes, usually 8 and 9 or 98 and 99 (see Appendix I of reference 2, in Appendix A and Table 3-2).

The SPSS system must be informed which values correspond to missing data since they require special handling. The SPSS instructions used to do this are MISSING VALUES and ASSIGN MISSING (see pp. 57-59 and 115-119, Reference 6, Appendix A). Of the new variables created by modifying survey questions, all those which have missing data use -1 as the missing data value.

3.4.3.1.6 SPSS Parameter Definitions

In this section:

- The printout format of certain variables is defined to allow digits to be printed after the decimal point
- A unit to which to write the z-scores (unit 20) is defined

The SPSS instructions used are PRINT FORMATS and RAW OUTPUT UNIT (see pp. 61 and 69, Reference 6, Appendix A).

3.4.3.1.7 Z-score Production

Z-scores are produced by invoking the CONDESCRIPTIVE procedure in SPSS. and specifying OPTION 3 (see pp. 181-193, Reference 6, Appendix A). They are written to unit 20 as specified by the RAW OUTPUT UNIT card. The z-scores are used by RECODE2 to compute compound variables. Subsection 3.4.4 describes the format in which they are written.

3.4.3.1.8 System File Creation

At the end of the RECODE1 program, all the data for all the variables, old and new, their labels and other information used by SPSS is written to unit 4. The SPSS instruction is SAVE FILE (see pp. 81 and 617, Reference 6, Appendix A). The z-scores are not included in this file. The file is labeled internally as SYSFILE1. This internal label does not have to correspond to the name by which the file is known to the EXEC 8 operating system.

3.4.3.2 Constraints/Limitations

RECODE1 cannot detect erroneous data. If the data have been improperly cleaned then incorrect results will be generated.

3.4.3.3 Execution Time Requirements

To process 700 cases of input data, RECODE1 requires 64 kWords of main memory and 2:15 SUPS.

3.4.3.4 Peripheral Storage Requirements

Assuming 700 cases of survey data, RECODE1 requires:

- disk tracks for the system file it creates (proj-id*SYSFILEL)
- disk tracks for the z-scores file it creates (proj-id*ZSCORES)

3.4.3.5 Differences between Forms of RECODE1

RECODE1 has two forms:

- One used for communities that have a noise control program
- One used for communities that do not have a noise control program

Whether or not a community is performing acoustical monitoring has no effect on RECODE1.

The two forms differ in that:

- The variables RQ18A through RQ18F, RQ19AB through RQ19AD and Q19ADUMY, their labels and their missing values, are not defined for communities <u>with</u> noise control programs
- The labels for Q19, Q19AA, RQ19 and RQ19AA are changed

3.4.4 RECODE1 OUTPUTS

RECODE1 outputs two files:

- An SPSS system file, proj-id*SYSFILE1., containing the data values for all variables read or created by RECODE1, their labels, and other information of use to SPSS
- A file, proj-id*ZSCORES. containing z-scores for nine variables (see Figure 3-8)
- A printout of frequency distributions from the CONDESCRIPTIVE procedure.

Both of the files are created for later use by the RECODE2 program (see 3.5 below). The first file is in standard SPSS system file format (see Chapter 7, Reference 6, Appendix A). The z-score file is a FORTRAN file in the format used by the CONDESCRIPTIVE procedure of SPSS. In this case, this produces a file with the format:

```
N(16X,8F8.5/16X,1F8.5)
```

where N is the number of cases read.

Storage requirements are dealt with in 3.4.3.4 above.

3.4.5 INTERFACE REQUIREMENTS

Two other processes are required to have been completed previous to RECODE1's execution: .data cleaning .TAILOR.

The data cleaning process, described in 3.1, produces the data on which RECODE1 operates.

The TAILOR program described in 3.3, creates RECODE1 itself.

Q12
Q13A
Q13B
Q13C
Q19
Q20
Q21SUM

FIGURE 3-8. Variables for which RECODE1 Calculates z-scores

3.4.6 PROGRAM VARIABLE DESCRIPTION

RECODE1 has two types of variable:

- Unmodified survey variables
- Modified variables created from survey variables

The first type is given the same names in RECODE1 as they are given in Section 5.2.1 of Reference 1, Appendix A. For almost all, this means that the first letter of the variable name is a 'Q', though for nine variables mnemonic names are used (for those variables in which both 'Q' names and mnemonic names are given by reference 1, the 'Q' names are used).

The variables created by RECODE1 retain the previous names with prefixes added according to the following conventions:

R = Missing values recoded with no other modification
RM = Missing values recoded with slight modification to other values
RT = Missing values recoded and variable values grouped
RTH = Missing values recoded and variable values converted to a Thurstone Scale
RF = Missing values recoded and other recodes performed to make values suitable for frequencies
RPG = Missing values recoded and other recodes performed to make values suitable for regressions

Other variables created by RECODE1 are given mnemonic names as shown in Figure 3-9.

3.4.7 PROGRAM EXECUTION REQUIREMENTS

Input and output files must be assigned @USE statements must be utilized to associate files with internal file names as follows:

File	Internal Name
Input file	8
System File	3
Z-scores file	20

SPSS version 7.2 must be used to interpret the statements in RECODE1.

Mnemonic Name	Created From
YRSRES	RMQ1
EVALAREA	Q2
NOISPROB	Q5C
NSECONTL	Q6K
HEALTH	Q7
QUIET	RQ8
ANNOY	RTHQ9
TRAFFIC	RTHQ17A
MTRCYCLE	" B
TRUCKS	" C
BUSES	" D
AUTOS	" E
HIGHWAY	" F
RECRLVHC	" G
GRBGETRK	" H
EMERGVHC	" I
ENTMTCTR	" J
PETS	" K
NBRHOMES	" L
AIRCONDT	" M
GRDNEQPT	" N
JETS	" O
SMLPLANE	" P
HCOPTERS	" Q
TRAINS	" R
CONSTRCN	" S
COMINDUS	" T
AGE	RQ22
EDUCLVL	RQ23
INCOME	RRGQ26
OWNER	RQ28
SEX	RQB1
STRDIST	QB3

FIGURE 3-9. Mnemonically Named Variables Created in RECODE1

3.5 The RECODE2 Program

This subsection describes the program RECODE2/V1.0(0). RECODE2 is written in SPSS.

3.5.1 SOFTWARE FUNCTION

RECODE2 is needed to deal with the following circumstances:

- The attitudinal survey analysis schema (see Chapter 5 of Reference 1, Appendix A) calls for the creation of several compound variables
- Some of these compounds must be created using z-scores or standardized scores, of other variables
- SPSS will calculate z-scores, but they are written to disc and cannot be accessed by the program which calculates them

Since using a second recoding program was inescapable, it was decided to define all compound variables in the second program, including those that could have been calculated in the first. This makes the system easier to follow by making these calculations easier to find.

Definition of the variables DBLEVEL and (when appropriate) DBLVLSQR also occurs in RECODE2.

3.5.2 INPUTS

RECODE2 reads two files for input:

- proj-id*SYSFILE1. described in 3.4.4 above
- proj-id*ZSCORES. also described in 3.4.4

Both files are created by the RECODE1 program.

3.5.3 PROCESSING

Because slightly differing forms of RECODE2 must be used to process data from different cities, RECODE2 is modified by TAILOR (See 3.3 above). Four basic forms can be produced. Since two of these forms make use of community collected acoustical data which is unique for each city, these forms of RECODE2 are also unique. Differences between forms are described in 3.4.3.4 below.

3.5.3.1 Program Flow

The logical flow of RECODE2 is shown in Figure 3 - 10. Table 3 - 4 lists the significant variables created by RECODE2. The steps taken in RECODE2 are very similar to those in RECODE1 and so most will not be described in detail. The reader is referred to 3.4.3.1 above for such a description.

Three points are worth noting:

- The method of defining the variable SUPPORT
- The method of defining the variable DBLEVEL
- The reason for calling the FREQUENCIES procedure

The first two points are covered in 3.4.3.5 below. The FREQUENCIES procedure was invoked only to force SPSS to read the data and perform the data manipulations specified earlier in the program.

3.5.3.2 Constraints/Limitations

RECODE2 is vulnerable to a user mistake in the definition of DBLEVEL. This mistake would occur when the user runs TAILOR on RECODE2 and could only happen if the community involved had a noise control program (See 3.3 and 3.5.3.5). If such a mistake were to go undetected, the results from the REGRESSIONS program would be in error as would some of the results from the FREQUENCIES and CROSSTABS programs.

There are two types of errors a user could commit:

- Errors in the SPSS control card format
- Errors in the content of the control cards

Format errors will be detected by SPSS when RECODE2 is run. The program will bomb with an error message indicating the incorrect statement.

Content errors are of two types:

- Incorrect decibel codes
- Incorrect cluster codes

216

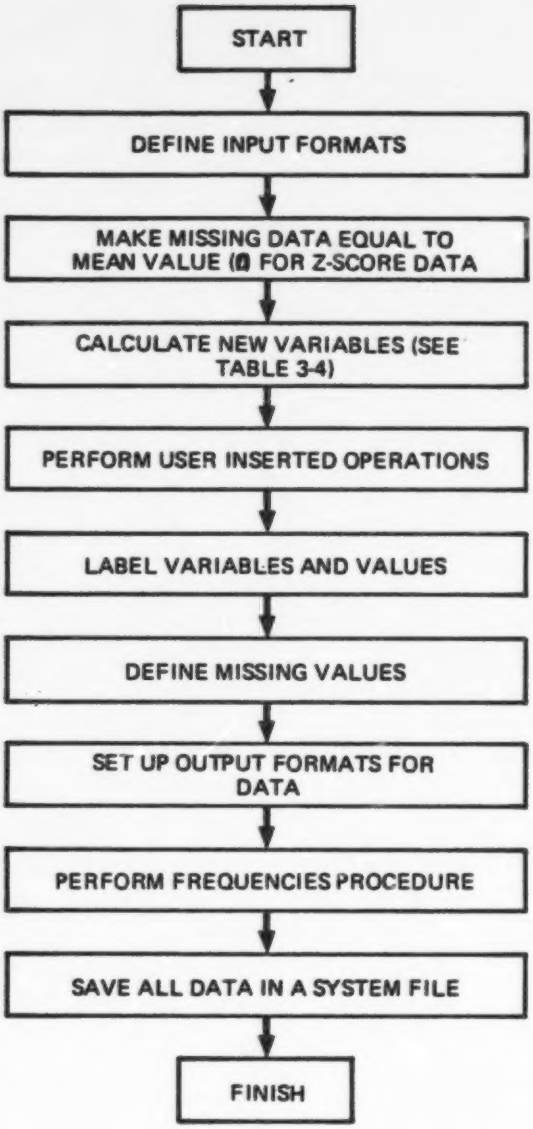

START
DEFINE INPUT FORMATS
MAKE MISSING DATA EQUAL TO MEAN VALUE (0 FOR Z-SCORE DATA
CALCULATE NEW VARIABLES (SEE TABLE 3-4)
PERFORM USER INSERTED OPERATIONS
LABEL VARIABLES AND VALUES
DEFINE MISSING VALUES
SET UP OUTPUT FORMATS FOR DATA
PERFORM FREQUENCIES PROCEDURE
SAVE ALL DATA IN A SYSTEM FILE
FINISH

Figure 3-10

RECODE2 Flow Chart

TABLE 3-4 Variables Created in RECODE2

New Variable	Variable from which created (Z suffix indicates Z-scores
URBPROB	NQ5A through NQ5H
PUBSERV	NQ6A through NQ6K
HIGHLY	Q9
ACTIVITY	NQ10A through NQ10D
HLTHPROB	Q12Z,Q13AZ through Q13EZ
PREVENT	NQ15A through NQ15E
COMPLNT	NQ16A through NQ16C
SUPPORT	Q19Z, Q20Z, Q21SUMZ (If community does not have a noise control program)
	Q20Z, Q21SUMZ (If community does have a noise control program)
DBLEVEL	NZONE (If community does not have an acoustical monitoring program)
	Acoustical measurements (If community does have an acoustical monitoring program)
DBLVLSQR (Created only if community has an acoustical monitoring program)	DBLEVEL

Decibel code errors can be further subdivided into:

- Those that remain in range (30 to 110 decibels)
- Those that are out of range

Recodes that assign incorrect but in-range values to DBLEVEL will not be detected by any program in the ASDAS. Out-of-range values for DBLEVEL will cause error messages to be generated by FREQUENCIES and CROSSTABS but not REGRESSIONS.

Errors in the cluster numbers will be detected by FREQUENCIES and CROSSTABS only if they result in a cluster number less than 30 remaining unrecoded (i.e., appearing to following programs as an out of range DBLEVEL). Otherwise they will remain hidden to all the following programs.

3.5.3.3 Execution Time Requirements

To process 700 cases of data, RECODE2 requires 64 kWords of main memory and 0:30 SUPS.

3.5.3.4 Peripheral Storage Requirements to Operation

To operate on 700 cases of survey data, RECODE2 requires:

- 100 tracks for the output file, proj-id*SYSFILE2. (See 3.5.4 below)

3.5.3.5 Differences between forms of RECODE2

The forms of RECODE2 differ in their handling of these variables:

SUPPORT	(Degree of community support for a noise control program)
DBLEVEL	(Physical noise level)
DBLVLSQR	(The square of DBLEVEL)

SUPPORT is defined differently according to whether or not a community has a noise control program.

SUPPORT = Q19Z+Q20Z+Q21SUMZ

219

Where Q19Z, Q20Z and Q21SUMZ are the z-scores for questions 19, 20 and Q21SUM=Q21A+Q21B+....Q21H.

If there is a noise control program:

SUPPORT = Q20Z+Q21SUMZ

Since question 19 no longer has any relevance to SUPPORT.

For those cities which do not have acoustical monitoring data, the variables DBLEVEL AND DBLVSQR are defined as follows:

DBLEVEL =	NZONE, where NZONE is noise zone number (see chapter 3 of reference 1 Appendix A)
DBLVLSQR	is undefined

For cities which do have acoustical monitoring data, the variables are defined as follows:

COMPUTE DBLEVEL=CL

(where CL is the same as CLUSTER) sets DBLEVEL equal to the CLUSTER the respondent lives in. Then the decibel levels measured within the cluster is substituted for each cluster number with the statement:

RECODE DBLEVEL $(1=dB_1)$ $(2=dB_2)$..,

.

.

.

$(100=dB_{100})$

Where dB is replaced by the cluster's noise level in decibels. Finally DBLVLSQR is defined by:

COMPUTE DBLVLSQR=DBLEVEL*DBLEVEL

The variations in the definitions of SUPPORT, DBLEVEL and DBLVLSQR require small variations in the labeling and FREQUENCIES portions of RECODE2.

3.5.4 OUTPUTS

RECODE2 produces the file proj-id*SYSFILE2. It is in SPSS system file format (See Chapter 7, Reference 6, Appendix A). It contains all the data in proj-id*SYSFILE1. plus all variables read or created by RECODE2.

3.5.5 INTERFACE REQUIREMENTS

Two other programs must be successfully run before RECODE2:

- TAILOR
- RECODE1

TAILOR creates the correct version of RECODE2 for the community using the ASDAS.

RECODE1 creates the data file needed for input to RECODE2.

3.5.6 PROGRAM VARIABLE DESCRIPTIONS

RECODE2 has three types of variables:

- Old variables from proj-id*SYSFILE1.
- Z-scores
- New variables created within RECODE2

The old variables retain their names. Their values are also unaltered.

The z-scores are variables created by RECODE1 from certain of the old variables mentioned above. (The z-scores are given the names of the variables from which they were created, but a suffix of 'z' is added to differentiate them.

The variables created within RECODE2 are given the mnemonic names suggested on Pages 5-8 of Reference 1, Appendix A.

3.5.7 PROGRAM EXECUTION REQUIREMENTS

Input and output files must be assigned. @USE statements must be utilized to associate internal names with files as follows:

File	Internal Name
System input file	4
Z-scores file	8
System output file	3

SPSS version 7.2 must be used to interpret the statements in RECODE2.

3.6 THE FREQUENCIES, CROSSTABS, AND REGRESSIONS PROGRAMS

This subsection describes the FREQUENCIES V1.0(0), CROSSTABS V1.0(0) and REGRESSIONS V1.0(0) programs. These programs are written in SPSS.

These three programs are very similar in construction and usage, despite having very different functions.

3.6.1 SOFTWARE FUNCTIONS

Each program has a different purpose:

FREQUENCIES calculates frequency distributions for most of the variables in the questionnaire. It also calculates:

- Minimum
- Maximum
- Range and
- Mode

statistics for all variables and

- Mean
- Median
- Variance
- Standard error
- Kurtosus and
- Skewness

for those variables for which this is appropriate.

CROSSTABS produces crosstabulations or contingency tables for many pairs of the survey's variables. It also calculates the chi square statistic for each pair.

REGRESSIONS determines the coefficients of several equations. These equations are those prescribed by the causal path model discussed in Chapter 5 of Reference 1, Appendix Z. It calculates the mean, standard deviation, and number of valid cases for each variable entered into the equations. Pearson product moment correlations between all the variables used in the regression equations are also calculated.

222

3.6.2 PROCESSING

3.6.2.1 Program Flow

The flow of each of these three programs is very straightforward. The programs can serve as their own flow charts. In general, they

- Read the system file (proj-id*SYSFILE2. if the system has been run according to the User's Manual)
- Perform their function
- Finish

Noteworthy points about the program are:

FREQUENCIES:

- The SPSS procedures FREQUENCIES and MULT RESPONSE are used to produce the frequency tables (See pp. 194-202 of reference 6, Appendix A and pp. 3-15 of Reference 7, Appendix A)
- Frequencies tables for Q22, RMQ1, CMPDAY and TIME are printed in condensed form, in order to present them on a single page

CROSSTABS:

- The CROSSTABS and MULT RESPONSE procedures of SPSS are used to produce the crosstabulations. (See Chapter 16 of Reference 6 and pp. 3-15 of Reference 7, Appendix A)

REGRESSIONS:

- REGRESSIONS uses the SPSS procedures REGRESSION and PEARSON CORR. (See Chapters 20 and 21, Reference 6 and Chapter 18, Reference 6, Appendix A)
- The values:

  N=80
  F=0.010
  T=0.001

RESID=0.25, are used for these parameters in the Regression Procedure.

- A two-tailed significance test, is used in the PEARSON CORR procedure

3.6.2.2 Constraints/Limitations

FREQUENCIES and CROSSTABS both require:

30 DBLEVEL 110 with acoustical data

0 DBLEVEL 99 without acoustical data

REGRESSIONS has no such constraint.

The REGRESSIONS program is limited in the number of cases it can process. Currently the limit is 740 cases. If more cases must be analyzed, the user must request more memory, both on the @RUN card and on the ALLOCATE card within the REGRESSIONS program itself.

3.6.2.3 Execution Time Requirements

For 700 cases of data:

FREQUENCIES requires 64 kWords of main memory and 5:00 SUPS

CROSSTABS requires 64 kWords of memory and 4:00 SUPS

REGRESSIONS requires 64 kWords of memory and 3:00 SUPS

3.6.2.4 Peripheral Storage Requirements

FREQUENCIES, CROSSTABS, and REGRESSIONS require the following storage if their output is to be kept on disk:

• FREQUENCIES:	21	tracks (proj-id*FREQOUT.)
• CROSSTABS:	56	tracks (proj-id*XTABSOUT.)
• REGRESSIONS:	90	tracks (proj-id*REGRESSOUT.)

3.6.2.5 Differences between forms of FREQUENCIES, CROSSTABS and REGRESSIONS

Communities without acoustical data:

- FREQUENCIES:
  DBLEVEL is expected to be in the range of 0 to 99
- CROSSTABS:
  DBLEVEL is expected to be in the range 0 to 99
- REGRESSIONS:
  DBLVLSQR is not included in the regression equations or the computation of correlations.

Communities with acoustical data:

- FREQUENCIES:
  DBLEVEL is expected to be in the range 30 to 110
- CROSSTABS:
  DBLEVEL is expected to be in the range 30 to 110
- REGRESSIONS:
  DBLVLSQR is included in the regression equations and the computation of correlations

Communities without a noise control program:

- FREQUENCIES:
  Frequency distributions are calculated for these variables:
    - RQ18A to RQ18F
    - RQ19AA to RQAAD
    - Q19ADUMY
- CROSSTABS:
  No effect
- REGRESSIONS:
  No effect

Communities with a noise control program:

- FREQUENCIES:

  Frequency distributions are calculated for:

  RQ19AA
  but not for RQ18A to
  RQ18F, RQ19B to RQ19F
  or Q19DUMMY

- CROSSTABS:

  Crosstabulations are calculated for:

  RQ19AA by AREA
  RQ19AA by RQ25A
  RQ19AA by Q26

- REGRESSIONS:

  No differences

3.6.3 Outputs

FREQUENCIES, CROSSTABS and REGRESSIONS each write to internal file name 6, the line printer. In normal ASDAS use an alternate print file is set up to contain this output (See User's Manual). The three alternate print files are:

FREQUENCIES:	proj-id*FREQOUT.
CROSSTABS:	proj-id*XTABSOUT.
REGRESSIONS:	proj-id*REGRESSOUT.

They are in EXEC 8 PRINT$ format.

3.6.4 Interface Requirements

The program RECODE2 must be run to create the file proj-id*SYSFILE2, before FREQUENCIES, CROSSTABS and REGRESSIONS are run.

The TAILOR program must also have been run to create FREQUENCIES, CROSSTABS and REGRESSIONS.

3.6.5 DATA DESCRIPTION

See 3.4.6 and 3.5.6 above

3.6.6 PROGRAM EXECUTION REQUIREMENTS

proj*SYSFILE2. must be assigned and @USEed to internal file name 3. If it is desired to save the program's output in an alternate print file, this file must be set up.